

AURINKOPANEELEIDEN TUOTON SEURANTA

Infojärjestelmä

Petri Juntunen

Opinnäytetyö
Tekniikan ja liikenteen ala
Tietotekniikan koulutus
Insinööri (AMK)

2014

ALKUSANAT

Haluaisin kiittää kaikkia opinnäytetyön tekoon vaikuttaneita henkilöitä. Aila Petäjäjärveä, jolta sain aiheen työhöni sekä Tapani Ruokasta ja Tommi Faarista, joita sain ohjausta työni tekoon. 2.6.2014.

TIIVISTELMÄ

LAPIN AMMATTIKORKEAKOULU, Tekniikan toimiala

Koulutusohjelma:	Tietotekniikka
Opinnäytetyön tekijä(t):	Petri Juntunen
Opinnäytetyön nimi:	Aurinkopaneelien tuoton seuranta
Sivuja (joista liitesivuja):	73 (24)
Päiväys:	21.8.2014
Opinnäytetyön ohjaaja(t):	Tapani Ruokanen
<p>Tämä opinnäytetyö käsitteli aiheeltaan tekniikan yksikön aurinkopaneelien tuottaman tehon seurantaa ja niiden tuottamien tehotietojen siirtämistä graafisessa muodossa koulun infojärjestelmään sekä koulun internet-sivuille. Lisäksi työssä oli tarkoitus tutustua aurinkopaneelien sähköntuotantoon, aurinkosähköjärjestelmiin ja infojärjestelmään. Muita tehtäviä oli myös miettiä parempi sijoituspaikka infojärjestelmän serverille, keille sen salasanat annetaan, millaista tuotantokäyttöä sillä olisi ja miten infojärjestelmän tiedot voisi parhaiten varmuuskopioida. Viimeiseksi tehtäväksi jäi vielä laatia käyttöohje infojärjestelmän käyttäjille. Aiheen sain Lapin Ammattikorkeakoulun toimeksiantona ja tein työtä kesän ajan: aluksi touko-kesäkuun koululla ja heinä-elokuun kotona.</p> <p>Aurinkopaneelilta saatu informaatio välittyy srea:n kautta MySQL-tietokantaan, josta tiedot ovat haettavissa sähkötekniikan infojärjestelmään käyttäen tekemiäni PHP-pohjaisia sivuja. Nämä PHP-ohjelmat hakevat tietokannasta tiedot ja niiden pohjalta piirtää graafisen esityksen tai esittää muita aurinkopaneelisiin liittyviä tietoja. Koulun infojärjestelmässä voi määrittää, mitä esityssivuja näytetään ja muita siihen liittyviä seikkoja.</p> <p>Aineiston opinnäytetyöhöni keräsin aloituskeskusteluista, palavereista, erilaisista muistioista, itse asioita kokeilemalla ja paljon tietoa asiasta löytyi internetistä. Aloitin työn teon opiskelemalla php-ohjelmointia ja tietokantojen käyttöä. Sitten siirryin tekemään itse esityssivuja ja laitoin ne toimimaan infojärjestelmään. Tein kesän aikana tämän raportoinnin ja liitin siihen ohjeen InfoSign-ohjelman käytöstä.</p> <p>Työn tekeminen vaati ohjelmointia PHP:llä ja opiskelua koulun infojärjestelmästä, invertteristä, tietokannoista, srea:sta sekä aurinkopaneelien toiminnasta. Sain php-sivut toimimaan, ja esitys aurinkopaneelien tuoton seurannasta on nyt nähtävillä koulun käytävän monitorilla. Oppimista tapahtui koko opinnäytetyö prosessin ajan ja siitä on varmasti hyötyä tulevaisuudessa.</p>	
Asiasanat: Aurinkopaneeli, ohjelmointi, tietokanta	

ABSTRACT

LAPLAND UNIVERSITY OF APPLIED SCIENCES, Technology

Degree programme:	Information Technology
Author(s):	Petri Juntunen
Thesis title:	Monitoring of Solar Panel Production
Pages (of which appendixes):	73 (24)
Date:	27 August 2014
Thesis instructor(s):	Tapani Ruokanen
<p>This thesis dealt with the monitoring of the power production of the solar panels at the unit of Technology and the transfer of their production data in graphic form to the school's information system as well as the school's website. In addition, the objective of the study was to examine the electricity production of the solar panels, solar power systems, and school's information system. Another objectives were also to think about a better positioning of the information system server, to whom the server passwords and usernames shall be given, what kind of production use the solar system might have, and how the server information can best be backed up. The final objective was to draw up a manual information for the users of the system. The topic was assigned by the Lapland University of Applied Sciences and I did it during the summer, first in the school in May-June and the rest of summer in July-August at home.</p> <p>The solar panel information is transmitted to the MySQL database via srea adapter. In the database the data are searchable to the school's information system by using PHP pages which were created. These PHP applications seek information in the database and draw a graphical presentation from the data, or presents other information related to the solar panels. In the school's information system can be determined which pages are displayed and other related issues in the presentation.</p> <p>Materials were collected for the thesis at the opening discussions, a variety of memos, trying to do things by myself and a lot of information of the subject found on the Internet. The work was stated by studying PHP programming and databases. Then the presentation of the pages was created and made it work in the information system. This documentation was done during the summer and the instructions for using the information system was attached to it.</p> <p>When doing this work it required lot of programming with PHP and learning about the information system, inverter, databases, srea module and how solar panels work. Now the PHP pages work to and the presentation of the solar system production is now available to be seen on the school hallway monitor. Learning took place throughout the process, and it is certainly useful in the future.</p>	
Keywords: Solar panel, programming, database.	

SISÄLLYS

ALKUSANAT	2
TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	4
KÄYTETYT MERKIT JA LYHENTEET	7
1 JOHDANTO	8
2 AURINKOENERGIA TEORIAA	10
2.1 Aurinkoenergia	10
2.2 Auringonsäteilyn määrä Suomessa	10
2.3 Aurinkopaneelit	11
2.3.1 Aurinkokennojen toimintaperiaate	11
2.3.2 Aurinkopaneelin ominaiskäyrä	14
2.3.3 Aurinkopaneelien tehontuotto	15
2.3.4 Lämpötilan vaikutus paneelien tehontuottoon	15
2.3.5 Suuntauksen vaikutus paneelien tehontuotantoon	17
2.3.6 Siirtojohtimien aiheuttamat tehohäviöt	18
3 TEKNIIKAN YKSIKÖN AURINKOJÄRJESTELMÄN ESITTELY	19
3.1 Aurinkojärjestelmän osat	19
3.2 Tietokanta ja InfoSign	20
4 PHP-OHJELMOINNIN OSUUS	22
4.1 PHP-ohjelmointi	22
4.2 PHP-kielioppi	22
4.3 Muuttujat ja taulukot	23
4.4 Ohjausrakenteet ja ehtolauseet	23
4.5 Silmukat	24
4.6 MySQL-tietokanta kyselyt	24
4.7 pChart kirjastot	27
5 INFOJÄRJESTELMÄ	29
5.1 InfoSign	29
5.2 Käyttöohjeet	29
6 AURINKOPANEELEIDEN TEHON SEURANTA	35

6.1	Infojärjestelmän serveri	35
6.2	Varmuuskopiointi	35
6.3	PHP-sivut	36
6.3.1	Päiväkohtainen tehon seuranta	36
6.3.2	Päiväkohtainen energiantuotto	37
6.3.3	Viikkokohtainen energiantuotto	38
6.3.4	Kuukausittainen energiantuotto	39
6.3.5	12 kuukauden energian tuotot	40
6.3.6	Vuosien energiantuotot	41
6.3.7	Tilastosivut	42
6.3.8	Sääasema	44
6.4	Työn arviointi	46
7	POHDINTA	47
	LÄHTEET	48
	LIITTEET	49

KÄYTETYT MERKIT JA LYHENTEET

csv	comma-separated values on käytetty tiedostomuoto
KNX	Kenttäväyläprotokolla
IP	Internetprotokolla
php	Hypertext Preprocessor on ohjelmointikieli
srea	Remote monitoring adapter eli etävalvontasovitin
pChart	Esityksiin käytetty grafiikka kirjasto
MySQL	Tiedon tallennukseen käytetty tietokanta

1 JOHDANTO

Tämä opinnäytetyö käsittelee aiheeltaan Lapin ammattikorkeakoulun tekniikan alan Kemmin toimipisteen aurinkopaneelien tuottaman tehon seuranta ja niiden tuottamien teho-tietojen siirtämistä koulun infojärjestelmään sekä koulun nettisivuille. Aiheen sain Lapin ammattikorkeakoulun toimeksiantona, ja teen työtä kesän ajan: aluksi koululla ja loppu-kesällä teen kotoa opinnäytetyön dokumentoinnin. Esitän työni tulokset syksyllä koulujen alkaessa. Työn tekeminen vaatii ohjelmointia PHP:llä ja opiskelua koulun infojärjestelmästä, invertteristä, tietokannoista, srea:sta eli ABB etävalvontasovittimesta sekä aurinkopaneelien toiminnasta.

Sain tämän opinnäytetyön aiheen tiedustelemalla sitä Lapin ammattikorkeakoulun sähkötekniikan opettajalta Aila Petäjäjärveltä. Opinnäytetyöni liittyy aikoinaan aloittamaani tietotekniikan insinööri opintojen saamiseksi loppuun. Siis teen tämän opinnäytetyön vasta yhdistetylle Lapin ammattikorkeakoululle.

Tavoitteena on saada siis koulun aurinkopaneelien tuotot näkymään sähkötekniikan infojärjestelmässä ja internet-sivuilla. Sähkötekniikan internet-sivuille yhdistetään myös Rovaniemen kampuksen aurinkopaneelien sähköntuoton tiedot. Lisäksi työssä on tarkoitus tutustua aurinkopaneelien sähköntuotantoon, aurinkosähköjärjestelmiin ja infojärjestelmään. Muita tehtäviä on myös miettiä parempi sijoituspaikka infojärjestelmän serverille, keille sen salasanat annetaan, millaista tuotantokäyttöä sillä olisi ja miten infojärjestelmän sekä tietokannan tiedot voisi parhaiten varmuuskopioida. Viimeiseksi tehtäväksi tulee vielä laatia käyttöohje infojärjestelmän käyttäjille. Infojärjestelmässä on tarkoitus saada näkymään ja päivittymään sähköntuotot graafisesti vuorokausittain, viikoittain, kuukausittain ja vuoden aikana tuotetut sähköntuotot. Lisäksi muita srea:lta ja koulun sääasemalta saatavia tietoja voi myös esittää infojärjestelmässä.

Opinnäytetyöni auttaa saamaan aurinkopaneelien toiminnan näkyvään muotoon, niin että jokainen voi nähdä aurinkopaneelien tehon tuoton ja toiminnan käytännössä milloin vain. Esityksen aurinkopaneelien tehon seurannasta infojärjestelmässä ja koulunsivulla teen PHP-sivujen avulla ja niiden graafiseen toteuttamiseen käytän pChart -kirjastoja. Tuloksena esityksessä pitäisi näkyä aurinkopaneelien tehon ja energian tuotot graafisesti ja numerotietoina sekä yleistä tietoa niiden toiminnasta. Ensimmäinen tekemäni PHP-sivu

näyttää aurinkopaneelien vuorokauden aikana tuottaman tehon graafisena käyränä. Muut PHP-sivut esittävät aurinkopaneelien tuottaman energian määrää pylväsdiagrammeina eri ajanjaksoilta. Näitä seurattavia ajanjaksoja ovat vuorokauden, viikon, kuukauden ja vuoden energian tuotot. Muita esitettäviä tietoja ovat parhaat tuotantoajankohdat ja niiden määrät, paljonko sähköä paneelit ovat koko toiminta-aikana tuottaneet sekä kyseisen hetken tehon tuotto. Graafisessa esityksessä pystyosa esittää tehontuoton ja vaakataso esittää ajankulun, kuten tunnit, vuorokaudet tai kuukaudet.

Tarkoitus on rajata opinnäytetyö pääpiirteittäin siihen, että saan aurinkopaneelien tehot ja muut tiedot näkymään graafisesti koulun infojärjestelmässä ja koulunsivuilla. Yksi tehtävä on selvittää, miten saan koulun sääaseman mittaamat tiedot näkymään myös esityksessä. Lisäksi selvitän serveriin liittyvät seikat ja teen käyttöohjeen infojärjestelmästä.

2 AURINKOENERGIA TEORIAA

2.1 Aurinkoenergia

Aurinkosähkö on luonnollinen, uusiutuva ja päästötön energian muoto. Aurinko säteilyn kokonaisteho maanpinnalle on noin 170 000 terrawattia, mutta siitä ei voida nykytekniikalla hyödyntää kuin pieni osa. Maanpinnalle tuleva säteilyteho kattaisi ihmiskunnan sähköntarpeen 60 000 kertaisesti. Aurinkosähkö on siinä suhteessa puhdas energiamuoto, että sen tuotannossa syntyy jätettä tai päästöjä vain siihen liittyvien laitteiden valmistuksessa ja kierrätyksessä. (Finnwind.pdf 2013, 3.)

2.2 Auringonsäteilyn määrä Suomessa

Auringon kokonaissäteily koostuu sieltä suoraan tulevasta säteilystä ja hajasäteilystä, joka taas on ilmakehän ja pilvien heijastamaa tai maasta heijastuvaa säteilyä. Suomessa hajasäteilyn osuus kokonaissäteilystä on merkittävä johtuen maantieteellisestä sijainnistaan maanpallolla. Eteläisessä Suomessa jopa puolet vuoden säteilystä on hajasäteilyä. Aurinkopaneelien sähköntuoton kannalta ei ole käytännössä ole merkitystä, onko paneeleille tuleva säteily suoraa vai hajasäteilyä. Kuitenkin jos käytetään keskittäviä tai aurinkoa seuraavia järjestelmiä, on säteilyn oltava suoraa säteilyä, jotta tuotanto olisi tehokasta ja taloudellisesti järkevää. (Motiva www-sivut 2014, hakupäivä 9.6.2014.)

Aurinkopaneelien sijoittelu ja kallistuskulmat vaikuttavat suuresti niille tulevan kokonaissäteilyn määrään. Ympäristöstä ja erilaisista pinnoista heijastuva säteily voi lisätä paneeleille tulevaa säteilyä hetkellisesti jopa yli 20 %. Tällä on kuitenkin vuositasolla vaikutusta normaalisti muutaman prosentin luokkaa. (Motiva www-sivut 2014, hakupäivä 9.6.2014.)

Suomen olosuhteisiin aurinkoenergian käyttäminen sähköntuottoon soveltuu varsin hyvin. Aurinko säteilee vuodessa Etelä-Suomen korkeudella neliömetrin vaakapinnalle yhteensä noin tuhat kilowattituntia. Keski-Suomessa auringon säteilymäärä on noin 900 kilowattituntia ja Pohjois-Suomessa sen määrä on vielä 800 kilowattituntia vuodessa. On laskettu, että Suomessa saa aurinkopaneeleilla neliömetrin alalta noin 100 kilowattituntia

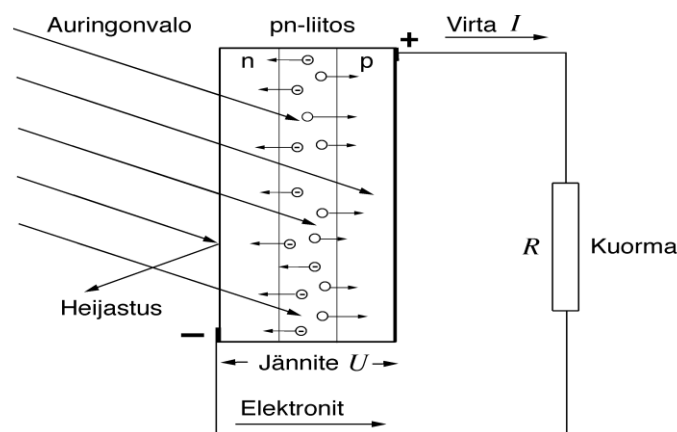
sähköä vuodessa. Aurinkosähkö sopii hyvin sähköverkon ulottumattomissa oleviin kohteisiin, koska se ovat edullinen ja huoleton vaihtoehto käyttösähkön tuottoon. (Finnwind.pdf 2013, 3.)

2.3 Aurinkopaneelit

Tässä kappaleessa käsittelen aurinkopaneelien ja aurinkokennojen toimintaperiaatteen sekä niihin liittyvää muuta teknistä tietoa. Tiedot teoriasta ja kuvat on suurimmaksi osaksi peräisin käyttämästäni Aurinkopaneelit.pdf -lähteestä.

2.3.1 Aurinkokennojen toimintaperiaate

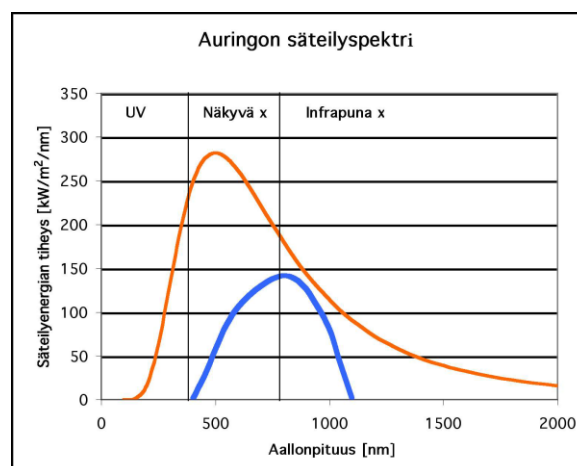
Aurinkokennon rakenne ja toimintaperiaate on melko yksinkertainen: kennoon tuleva auringon valo muuttuu siellä suoraan sähkövirraksi. Aurinkokenno on toimintaperiaatteenaan kuin suuri fotodiodi, jossa on yhdistetty kaksi erilaista puolijohdemateriaalia. Nämä puolijohdemateriaalit ovat nimetty p ja n merkinnöillä. Kun kennoon kohdistuu auringon valoa, niin osalla fotoneista eli valohiukkasista on niin suuri energia, että ne pääsevät ohuen pintakerroksen läpi pn-liitokseen ja voivat muodostaa elektroni-aukopareja. Lähellä pn-liitosta muodostuvista pareista elektronit kulkeutuvat n-puolelle ja aukot p-puolelle. Koska rajapintaan muodostuu sähkökenttä, voivat elektronit kulkea vain tiettyyn suuntaan. Elektronien on kuljettava ulkoisen johtimen kautta p-tyypin puolijohteeseen, jossa ne voivat yhdistyä sinne kulkeutuneiden aukkojen kanssa. Valaistun liitoksen eri puolilla on siten jatkuvasti vastakkaismerkkiset varauksenkuljettajat, ja liitos voi toimia ulkoisen piirin jännitelähteenä. (Suntekno Oy, Aurinkopaneelit.pdf.)



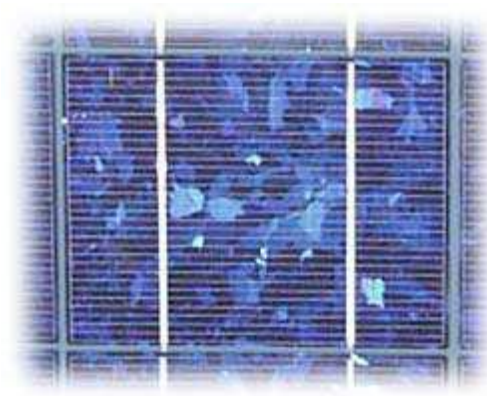
Kuva 1. Aurinkopaneelin toimintaperiaate. (Suntekno Oy, Aurinkopaneelit.pdf).

Pii on yleisin materiaali aurinkokennojen valmistuksessa, sitä käytetään yksi- ja monikiteisenä sekä myös amorfisessa muodossa. Kiteiset piikennot ovat yleensä noin 0,2 – 0,3 millimetriä paksuja ja pinta-ala vaihtelee (90 - 160) mm * (120 - 160) mm välillä. Yksikiteiset piikennot on sahattu yhtenäisistä piiaihioista, jonka halkaisija on 10 – 16 cm. Koska kennoihin käytettävä raaka-aine on hyvin kallista, ei pyöreistä kiekkoista kannata tehdä neliskulmaisia. Tästä johtuu yksikidepaneeleiden aukot kennojen kulmissa. Kuitenkin monikiteisiä piikennoja voidaan tehdä neliskulmaisista aihioista, jolloin raaka-aine voidaan käyttää tarkemmin hyödyksi. Amorfisesta piistä valmistetut kennot ovat taipuisia ja valmistuskustannuksiltaan halvempia, mutta niiden hyötysuhde on paljon pienempi. (Suntekno Oy, Aurinkopaneelit.pdf.)

Suurin aallonpituus, jolla fotoni saa aikaan elektroni-aukkoparin piissä on 1150 nm. Tällainen valo on lyhytaaltoista infrapunasäteilyä ja sen aallonpituus on lähellä näkyvän alueen rajaa. Jos säteilyn aallonpituus on suurempi kuin 1150 nm, se ainoastaan kuumentaa paneelia eikä synnytä sähkövirtaa. Jokainen fotoni voi synnyttää vain yhden elektroni-aukkoparin ja lisäksi jos fotoneilla on enemmän energiaa kuin tarvitaan aukkoparin synnyttämiseen, niin osa fotonien energiasta menee hukkaan. Piikennot eivät myöskään pysty hyödyntämään lyhytaaltoista ultraviolettivaloa, joka aiheuttaa paneelien tuhoutumisen pitkän ajan kuluessa. Alla olevassa kuvassa 2 on esitetty auringon säteilyspektrin muoto ilman ilmakehän absorptiota sekä piikennon absorptioalue. (Suntekno Oy, Aurinkopaneelit.pdf.)



Kuva 2. Auringon säteilyspektrin muoto sekä piikennon absorptioalue (sininen käyrä). (Suntekno Oy, Aurinkopaneelit.pdf.)



Kuva 3. Monikiteinen piikkenno. (Suntekno Oy, Aurinkopaneelit.pdf).



Kuva 4. Yksikiteinen piikkenno. (Suntekno Oy, Aurinkopaneelit.pdf).

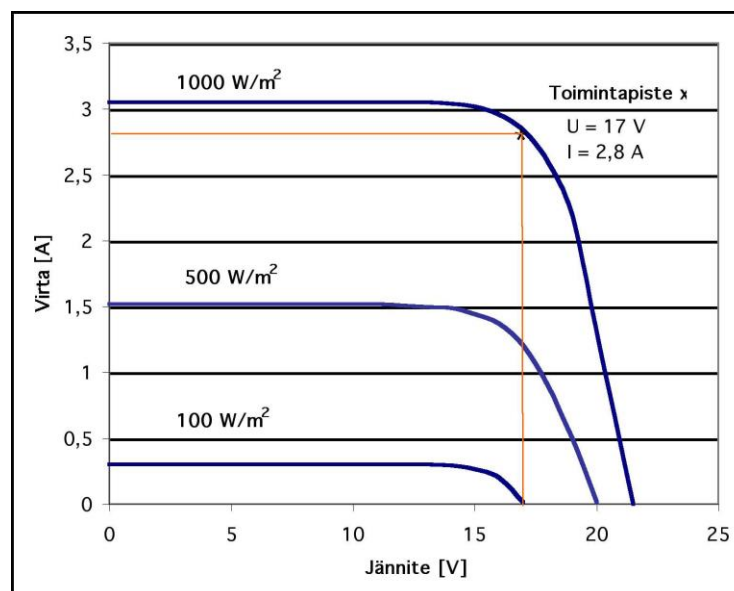
Monikiteisen ja yksikiteisen piikennon rakenne näkyy yllä olevista kuvista. Monikiteisen piikennon kuvasta näkyvät yksittäiset piikiteet sekä kennon pinnalle juotetut johtimet, joiden avulla syntynyt sähkövirta johdetaan ulkoiseen kuormaan. Johtimien kohdat eivät voi tuottaa sähköä. (Suntekno Oy, Aurinkopaneelit.pdf.)

Piikidekennojen teoreettiseksi hyötysuhteeksi on laskettu noin 31 %. Kennojen hyötysuhdetta heikentävät muun muassa metallijohteiden liitokset paneelien pinnalla, resistanssi sekä heijastukset paneelin päällä olevasta lasista. Lasista johtuvaa heijastusta voi vähentää käyttämällä erilaisia pinnoitteita ja näin parantaa kennojen hyötysuhdetta. Pinnoitteet tummentavat lasin pintaa ja siten vähentävät auringon valon heijastusta. Nykyään parhaiden piistä valmistettujen aurinkopaneelien hyötysuhde on noin 18 %. (Suntekno Oy, Aurinkopaneelit.pdf.)

Yhden aurinkokennon antama jännite on noin 0,5 – 0,6 V. Käyttötarkoituksensa mukaan ne kytketään sarjaan paneeleiksi. Yleensä paneelissa on 36 kennoa, jolloin saadaan nimellisarvoltaan 12 V jännite. Aurinkopaneeleista saatu sähkövirta on suoraan verrannollinen muodostuvien elektroni-aukkoparien lukumäärään. Tästä johtuu että sähkövirta riippuu kennon pinta-alasta ja auringon säteilyn voimakkuudesta. Kennot tuottavat kirkkaalla auringonpaisteella sähkövirtaa noin 32 mA/cm^2 . Siten esimerkiksi $90 \text{ mm} \times 120 \text{ mm}$ suuruinen kenno tuottaa enintään 3,5 A ja jos kennot ovat kytketty sarjaan, on paneelistä saatava virta yhtä suuri kuin yhden kennon tuottama virta. (Suntekno Oy, Aurinkopaneelit.pdf.)

2.3.2 Aurinkopaneelin ominaiskäyrä

Aurinkopaneelin ominaiskäyrä ilmoittaa, millä virran ja jännitteen arvoilla se voi toimia. Oikosulkuvirta on paneelin tuottama enimmäisvirta, kun sen navat on kytketty oikosulkuun. Tyhjäkäyntijännite on paneelin suurin jännite, kun paneeliin ei ole kytketty kuormaa. Maksimitehopiste tai toimintapiste on virran ja jännitteen arvot, joilla saavutetaan suurin ulostuloteho kyseisissä käyttöolosuhteissa. Käytännössä toimintapistettä ei voi saavuttaa, koska valaistusolosuhteet vaihtelevat ja paneelin lämpeneminen pienentää paneelin tehoa. (Suntekno Oy, Aurinkopaneelit.pdf.)



Kuva 5. 50 Wp paneelin ominaiskäyrä eri säteilyvoimakkuuksilla lämpötilassa 25 °C. (Suntekno Oy, Aurinkopaneelit.pdf).

Yllä olevasta kuvasta näkyy miten paneelin tuottama virta pienenee lähes suorassa suhteessa säteilyn määrään. Kuvasta näkyy myös miten jännite pienenee säteilyn vähetessä. Maksimitehopiste saavutetaan hieman sen jälkeen, kun virta on käyrällä alkanut laskea. (Suntekno Oy, Aurinkopaneelit.pdf.)

2.3.3 Aurinkopaneelien tehontuotto

Aurinkopaneeleista saatava teho lasketaan Joulen lakiin perustuvasta kaavasta $P = UI$, missä P on teho (W), U on jännite (V) ja I on virta (A). Paneelin tuottama energia saadaan kaavasta $E = Pt$, kun teho kerrotaan ajalla. Yleensä sähköenergia annetaan yksiköissä wattitunti (Wh) tai kilowattitunti (kWh). Tällöin aika annetaan tunteina (h). (Suntekno Oy, Aurinkopaneelit.pdf.)

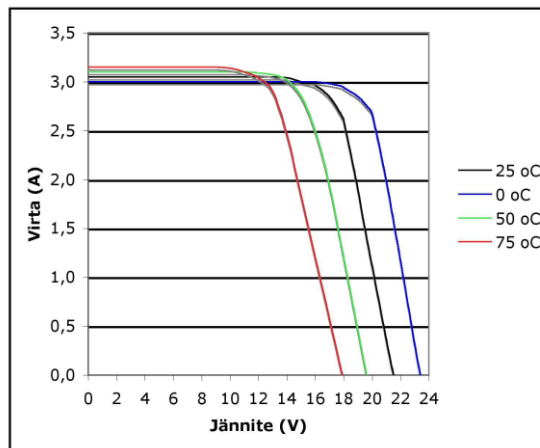
Paneelin hyötysuhde η saadaan paneelin tuottaman tehon ja paneelille tulevan säteilyn suhteena $\eta = \frac{P}{S_A} * 100 \%$, missä P on paneelin teho, S on auringonsäteilyn voimakkuus ja A paneelin pinta-ala. (Suntekno Oy, Aurinkopaneelit.pdf.)

Paneeliin kytketty kuorma tai akusto määrää paneelin jännitteen, jota vastaavaan pisteeseen virta hakeutuu kulloistakin säteilyä ja lämpötilaa vastaavalla ominaiskäyrällä. Kuorman suuruus voidaan laskea ohmin laista $R = \frac{U}{I}$, missä R on kuorman resistanssi (Ω), U on paneelin napajännite ja I on virta. (Suntekno Oy, Aurinkopaneelit.pdf.)

Aurinkopaneelien tehot annetaan tietyissä olosuhteissa, jotta niitä voidaan vertailla keskenään. Nämä standardiolosuhteet eli englanniksi Standard Test Conditions (STC) ovat: auringon säteilyn voimakkuus ja se määritellään 1000 W/m^2 , paneelin lämpötilaksi määritellään $+25 \text{ }^\circ\text{C}$ ja auringon spektri normitettu ilmassalle 1,5. Standardiolosuhteissa valmistajat määrittelevät tuottamiensa paneelien nimellistehon W_p eli Watts peak. (Suntekno Oy, Aurinkopaneelit.pdf.)

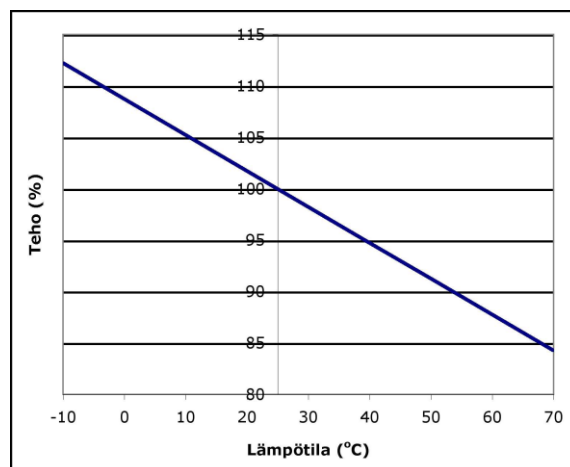
2.3.4 Lämpötilan vaikutus paneelien tehontuottoon

Aurinkopaneelin teho riippuu hyvin monista tekijöistä, kuten auringonsäteilyn voimakkuudesta ja tulokulmasta, auringon korkeudesta ja ilmakehän absorptiosta sekä paneelin lämpötilasta. (Suntekno Oy, Aurinkopaneelit.pdf.)



Kuva 6. Lämpötilan vaikutus paneelin ominaiskäyrään. (Suntekno Oy, Aurinkopaneelit.pdf).

Edellä olevasta kuvasta selviää aurinkopaneelin ominaiskäyrän muuttuminen eri lämpötilassa. Paneelin virta kasvaa sen lämpötilan noustessa, koska lämpötila lisää termisten varaustenkuljettajien määrää paneelissa. Tämä vaikutus on kuitenkin mitättömän pieni noin $+0,065 \text{ } \%/^{\circ}\text{C}$. Sen sijaan lämpötilan nousu pudottaa huomattavasti paneelin tyhjäkäyntijännitettä. Piikkenojen jännitteen lasku on yleensä noin $-0,5 \text{ } \%/^{\circ}\text{C}$ ja parhaimmillaan kennoilla jännitteen alenema on noin $-0,35 \text{ } \%/^{\circ}\text{C}$. Koska jännitteen muutos on paljon voimakkaampi kuin virran muutos, on tehon alenema lämpötilan noustessa samaa luokkaa kuin jännitteen lasku. (Suntekno Oy, Aurinkopaneelit.pdf.)



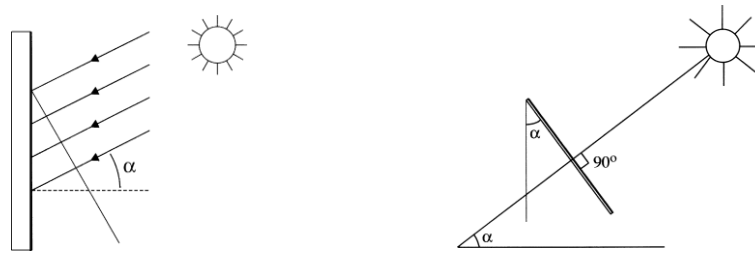
Kuva 7. Aurinkopaneelin tehon riippuvuus lämpötilasta. (Suntekno Oy, Aurinkopaneelit.pdf).

Ylläolevassa kuvasta näkyy miten paneelin lämpötila vaikuttaa sen tuottamaan tehoon, kun tehon lämpötilakerroin on $-0,35 \text{ } \%/^{\circ}\text{C}$. Normiteho 100 % on kiinnitetty lämpötilaan

+25 C°. Lämpötilalla on suuri merkitys tehon tuottoon, siksi onkin tärkeää sijoittaa paneelit siten, että tuuli ja ilmavirtaukset pääsisivät jäähdyttämään niitä. (Suntekno Oy, Aurinkopaneelit.pdf.)

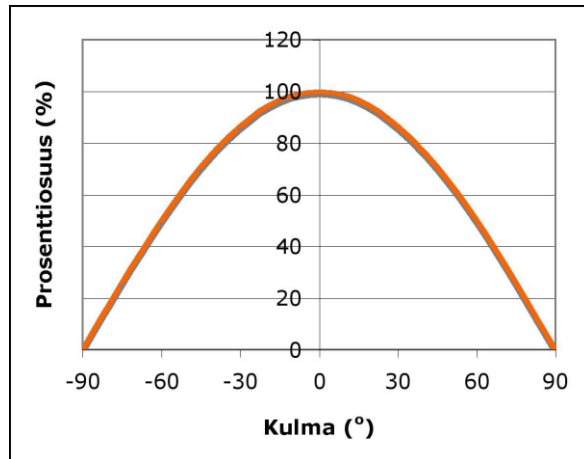
2.3.5 Suuntauksen vaikutus paneelien tehontuotantoon

Auringon säteily tulee kohtisuorasti paneelin pintaan kun sen kallistuskulma on sama kuin auringon korkeus horisontista. Auringon korkeus kuitenkin vaihtelee jatkuvasti päivittäin ja myös vuodenaikojen mukaan. Optimikulma Suomessa kiinteästi asennetuille paneeleille on suoran säteilyn kannalta 30 – 40°. Jos käytössä on aurinkoa seuraava järjestelmä voi energiaa saada 30 % enemmän kuin kiinteällä paneelilla. Pilvinen päivä ei tosin hyötyä tuo tässä tapauksessa. Alla olevasta kuvasta näkyy miten paneelille tuleva auringonsäteily ja paneelin asennuskulman pitäisi olla toisiinsa nähden. (Suntekno Oy, Aurinkopaneelit.pdf.)



Kuva 8. Paneelille tuleva auringonsäteily ja paneelin suuntaus aurinkoa kohti. (Suntekno Oy, Aurinkopaneelit.pdf).

Vinosti paneelin pinnalle tuleva auringonsäteilyn teho voidaan laskea helposti yhtälöstä $P_s = SA \cos a$, missä S on auringonsäteilyn voimakkuus (W/m^2), A on paneelin pinta-ala (m^2) ja a on paneelin normaalin ja auringonsäteiden välinen kulma. Huomattavaa on, että auringonsäteiden kohtaamiskulmaa on tarkasteltava sekä pystysuunnassa että vaakasuunnassa. Lisäksi kun auringonvalo tulee hyvin vinosti paneelin pintaan, eli tulokulma on yli 60°, niin heijastukset paneelin suojalasista tulevat merkittäviksi. Alla oleva kuva näyttää kohtaamiskulman vaikutuksen paneelille tulevaan auringonsäteilyn tehoon. (Suntekno Oy, Aurinkopaneelit.pdf.)



Kuva 9. Kohtaamiskulman vaikutus paneelille tulevaan auringonsäteilyn tehoon. (Suntekno Oy, Aurinkopaneelit.pdf).

2.3.6 Siirtojohtimien aiheuttamat tehohäviöt

Johtimissa kulkeva virta lämmittää johdinta ja aiheuttaa siten tehohäviöitä. Tehohäviöt voidaan laskea yhtälöstä $P = I^2 R$, missä I on johtimessa kulkeva virta ja R on johtimen resistanssi. Siirtohäviöiden pienentämiseksi johtimissa kulkeva virta tulisi olla mahdollisimman pieni. Myös johtimen resistanssi tulisi olla mahdollisimman pieni, eli johtimen pitäisi olla paksu ja hyvin sähköä johtavaa. Kuparia käytetään yleisesti sähkönjohtimissa ja hopeaa aurinkopaneeleiden pintaliitoksissa, koska näillä on parhaat sähkönjohtokyvyt. Yleensä johtimet mitoitetetaan siten, että häviöt eivät ylitä 5 %. Mitä pidempi johdin, sitä paksumpi sen pitää olla, jotta tehohäviöt eivät kasva liian suuriksi. (Suntekno Oy, Aurinkopaneelit.pdf.)

3 TEKNIKAN YKSIKÖN AURINKOJÄRJESTELMÄN ESITTELY

Lapin ammattikorkeakoulun Kemin tekniikan yksikön aurinkopaneelit ovat BYD nimi-sen yrityksen valmistamat. Ne on asennettu koulunkatolle ABB:n toimesta, kuten muutkin aurinkosähköjärjestelmään kuuluvat laitteet. Tähän opinnäytetyöhön liittyviä laitteita ovat itse aurinkopaneelit, invertteri, SREA-50 sovitin, infojärjestelmän osat ja sääasema. Lähteinä tässä kappaleessa on monia ja niistä olen koostanut seuraavan tekstin. Tiedonlähteitä ovat projektinaloituksessa saamani dokumentti ja keskustelut asiaan liittyvien henkilöiden kanssa, sekä Moodlesta löytyvä kurssin 3S2586_työelämäprojekti_1_k2014 aurinkopaneelijärjestelmä työ 11.

3.1 Aurinkojärjestelmän osat

BYD:n valmistamia 240 W aurinkopaneeleita on kytketty yhteen 14 kappaletta ja niistä saadaan 3,36 kW_p kokonaisteho. Aurinkopaneeleilta on vedetty 4 mm² DC-kaapelointi invertterille. Pinta-alaa aurinkopaneeleilla on yhteensä 22,8 m² ja painoa on 265 kg.



Kuva 10. PVS300 invertteri ja SREA-50 adapteri (ABB:n internet-sivut 2014, ABB PVS300 string inverter manual.pdf ja ABB SREA-50 remote monitor adapter.pdf).

Invertterinä toimii ABB PVS300-TL-3300W-2 ja sen tarkoitus on muuntaa aurinkopaneeleita tuleva tasavirta vaihtovirraksi syötettäväksi eteenpäin. Siihen on liitetty etäkäytettävä tarkkailuadapteri SREA-50, jolla seurataan aurinkopaneeleilta invertterille tulevaa tietoa. ABB SREA -ohjelmalla pystyy näkemään kaiken tiedon milloin vain ja tiedot voi lähettää edelleen internetin välityksellä esimerkiksi käyttämäni tietokantaan csv-

tiedostona. Ohjelmistossa voi määrittellä, kuinka kauan tietoja säilytään, tietojen päivitysvälin pituus ja mihin osoitteeseen tiedot välitetään. Näillä tiedoilla on siis merkitystä opinnäytetyössäni.

Alla olevan kuvan sääasema on asennettu kiinteistösähköistyslaboratorion ulkoseinälle. Sääasema on yhdistelmäyksikkö, jota voidaan käyttää tuulen, sateen, kirkkauden ja lämpötilan tunnistimena kiinteistöissä. Itse tunnistinarvot ja mittausarvot käsitellään laitteessa. Sääasemaa käytetään KNX-väylän kanssa, johon mittausarvot lähetetään suoraan.



Kuva 11. Sääasema. (Schneider-electric [www-sivut](http://www.schneider-electric.com), KNX Basic weather station (SE_NO_FL_DA).pdf.)

3.2 Tietokanta ja InfoSign

MYSQL-tietokanta, johon SREA lähettää csv-tiedoston on nimeltään ”srea_data” ja siellä on taulukko ”pvs300”. Sinne tallentuu 10 minuutin välein päivämäärät, kellonajat, kasvava sekuntimäärä tammikuun alusta vuodesta 1970, tehon tuotto, parhaat tuotantomäärät ja ajat, järjestelmän käyttötunnit sekä järjestelmän kokonaistehontuotto. Erillistä tietokantaa joudutaan käyttämään esityksen perustana, koska SREA kirjaa käytetyn tunnuksen ulos automaattisesti maksimissaan 24 tunnin välein. Seuraavalla sivulla on taulukko csv-tiedoston kentistä ja siitä minkä tyyppisiä ne ovat.

Taulukko 1. CSV-tiedoston kentät ja data-tyypit

Date	2014-03-18	(DATE)
Time	08:30:00	(TIME)
Seconds from 1 jan 1970 (GMT)	1395124200	(INT)
Power [kW]	1.74	DOUBLE(6,2)
Power peak of day [kW]	1.74	DOUBLE(6,2)
Power peak time of day (Unix epoch) [s]	1395124196.00	(INT)
Power peak ever [kW]	3.45	DOUBLE(6,2)
Power peak time ever (Unix epoch) [s]	1394270781.00	(INT)
Total energy [kWh]	165.35	DOUBLE(10,2)
Total feed [hour.hour/100]	366.02	DOUBLE(10,2)

Sähkötekniikan infojärjestelmään kuuluu näyttö koulun käytävällä, serverinä toimiva tietokone ja siinä toimiva InfoSign-ohjelma. Aurinkopaneelien tiedot näkyvät käytävän näytöllä, jota ohjataan siis sähkövoimatekniikanlaboratoriossa olevalla tietokoneella ja InfoSign-ohjelman avulla. Kyseinen ohjelma pyörittää jatkuvaa esitystä, joka koostuu tekemistäni php-sivuista ja powerpoint-esityksestä. Tekemäni php-sivut käytännössä hakee tiedot MYSQL-tietokannasta ja niiden perusteella muodostaa graafiset kuvaajat näkyville. Tarkempi kuvaus InfoSign-ohjelmasta löytyy tekemästäni käyttöohjeesta.

4 PHP-OHJELMOINNIN OSUUS

4.1 PHP-ohjelmointi

Käyn tässä kappaleessa läpi opinnäytetyössäni tarvitsemani PHP-ohjelmointiin liittyvät seikat. Näitä tarpeellisia seikkoja ovat lähinnä kielioppi, eli syntaksi, ja ohjausrakenteet, kuten muuttujat, taulukot ja ehtolauseet. Lisäksi opiskelin MySQL-tietokantojen peruskäytön ja siitä lähinnä tietokannasta tehtävien kyselyjen muodostamisen. Itse työssäni ei tarvinnut normaalisti tehdä tai lisäillä mitään tietokantaan. Lisäksi opinnäytetyössäni tärkeä rooli oli pChart-kirjastolla, jota ilman ei graafista esitystä olisi saatu aikaan.

PHP on lyhenne sanoista Hypertext Preprocessor ja se on tarkoitettu lähinnä www-sovelusten ohjelmointiin. PHP on scriptikieli, joka muistuttaa syntaksiltaan Perl ja C-ohjelmointikieliä. Sen kehittämisen on aloittanut Rasmus Lerdorf vuonna 1994 ja se on tarkoitettu avoimen lähdekoodin ohjelmistoksi. Nykyinen käytössä oleva ohjelmiston versio on PHP 5. PHP on kehitetty alusta alkaen erityisesti web-ympäristön sovelluksia varten ja PHP-scriptit voi upottaa suoraan sivun HTML-koodin sekaan. Koodin käsittelyn hoitaa palvelimella niin sanottu PHP-tulkki, sivun pyynnön yhteydessä. Tämä PHP-tulkki on nykyään asennettu suureen osaan eri palveluntarjoajien web-palvelimista. (PHP-perusteet, hakupäivä 16.6.2014.)

Ohjelmointityössäni jouduin paljon käsittelemään aika- ja päivämäärätietoja. Enimmäkseen käytin niihin date() funktiota. Tälle funktiolle annetaan parametreina muotoiluvarvot ja niin sanottu aikaleima.

4.2 PHP-kielioppi

PHP-koodi lisätään HTML-koodin sekaan `<?php ?>` -elementin sisään, ja sivu tallennetaan .php tiedostopäätteellä. Rivit lopetetaan puolipisteellä, poikkeuksena koodin viimeinen rivi. Koska koodi suoritetaan palvelimella, eivät PHP-tagit ja niiden sisällä oleva koodi, näy selaimelle lähetetyllä sivulla. Sivu siis sisältää ainoastaan koodin perusteella muokatun HTML-koodin. (PHP-perusteet, hakupäivä 16.6.2014.)

Koodin sekaan voi lisätä kommentteja, joko yksirivisenä tai monirivisenä. Kommentoinnin tarkoitus on tehdä selvemmäksi mitä ohjelmoija koodillaan haluaa tehdä myös muille koodin muokkaajille. Kommentteja ei oteta huomioon koodin suorituksessa. Yksirivinen kommentti alkaa kahdella kauttaviivalla, joiden jälkeen tulee varsinainen kommenttiteksti. Monirivinen kommentti alkaa kauttaviiva-kertomerkki yhdistelmästä ja jatkuu kommenttina niin kauan, kunnes saavutetaan kertomerkki-kauttaviiva yhdistelmä. (PHP-perusteet, hakupäivä 16.6.2014.)

4.3 Muuttujat ja taulukot

Muuttuja on muistipaikka, johon voi tallentaa koodissa tarvittavaa tietoa. Muuttujalle on annettu nimi, jolla siihen voi viitata koodissa. Yleensä se sisältää merkkijonon tai jonkin lukuarvon. Lukuarvomuuttujilla voi myös suorittaa erilaisia laskutoimituksia, joilla saadaan haluttu lopputulos. Käytin omissa sivuissani muuttujia muun muassa erilaisissa laskureissa tai silmukkarakenteissa, erilaisten ajankohtien nimissä ja yleensäkin tietojen säilymisessä tai niiden esittämisessä. (Ohjelmointiputka, hakupäivä 17.6.2014.)

Jos koodissa alkaa olla paljon käsiteltävää tietoa, voi olla epäkäytännöllistä käyttää pelkkiä muuttujia. Tällöin on käytännöllisempää käyttää taulukoita, koska taulukkoon mahtuu suuria tietomääriä vain yhden muuttujanimen alle. Taulukon varsinainen etu on, että hakasulkujen sisällä oleva alkio kohta voi olla muuttujan arvo. Taulukon alkioita voi koodissa käsitellä tai lukea helposti silmukkarakenteiden avulla. Taulukoita itse käytin työssäni tallentaakseni pitkiä MySQL-tietokanta hakujen tuloksia niihin. Tavallisesti tallennettavat tiedot olivat tehontuottoja tai erilaisia ajankohtia. Taulukoista tiedot oli helppo käsitellä tai siirtää graafisen piirtämisen käytettävään \$mydata-muuttujaan. (Ohjelmointiputka, hakupäivä 17.6.2014.)

4.4 Ohjausrakenteet ja ehtolauseet

Yksi tärkeä osa koodissa on if-rakenteet ja else if-rakenteet. Näiden rakenteiden avulla koodin toiminnan saa riippumaan muuttujista. Näissä rakenteissa asetetaan koodin suoritukselle ehtoja, joissa on mukana muuttujien arvoja. Tarkoituksena on suorittaa if-rakenteen sisällä olevan koodin vain silloin, kun alussa oleva ehto pitää paikkaansa.

```
if (ehto) {  
    koodi  
}
```

Yllä oleva esimerkki on if-rakenteen yksinkertaisin muoto. Lisäämällä if-rakenteeseen else-osan voi ohjelma suorittaa koodin, kun ehto ei pidä paikkaansa. Näitä rakenteita käytin tekemissäni php-sivuissa jonkin verran. (Ohjelmointiputka, hakupäivä 18.6.2014.)

4.5 Silmukat

Silmukan tarkoitus on toistaa samaa koodia monta kertaa peräkkäin. PHP-ohjelmointikielessä on kaksi erilaista silmukkaa: on for-silmukka ja while-silmukka. Tavallisin silmukan käyttötarkoitus on käydä läpi tietyllä lukuvälillä olevat luvut. Esimerkiksi for-silmukan avulla voi tulostaa saman tekstin halutun määrän kertoja tai läpi käydä kaikki merkkijonon merkit. Omassa koodissani while-silmukkaa käytin tallentamaan tietokannasta hakemaani tietoa erilaisiin taulukoihin ja for-silmukkaa käytin erilaissa laskureissa. Alla on esimerkit for-silmukan ja while-silmukan yksinkertaisimmista muodoista. (Ohjelmointiputka, hakupäivä 18.6.2014.)

```
for (aloitus; ehto; muutos) {  
    koodi  
}
```

```
while (ehto) {  
    koodi  
}
```

While-silmukka eroaa for-silmukasta, siten että se jatkaa koodin toistoa niin kauan kuin haluttu ehto pitää paikkaansa (Ohjelmointiputka, hakupäivä 18.6.2014).

4.6 MySQL-tietokanta kyselyt

Tekemäni esityksen toimimiseen täytyy jatkuvasti tallentaa, hakea ja joskus muuttaa tietoa. Tähän on omiaan MySQL-tietokantajärjestelmä, sinne on luonteva tallentaa ja sieltä hakea jatkuvasti päivittyvää tietosisältöä. Itse en käytettyyn tietokantaan tavallisesti lisää tai muokkaa mitään, vaan tieto sinne tulee suoraan SREA:lta. Tekemäni ohjelmakoodit tekevät lähinnä kyselyjä tietokantaan ja noutavat sieltä tarvitsemansa tiedon esimerkiksi pylväsdiagrammin piirtoon. (Ohjelmointiputka, hakupäivä 19.6.2014.)

MySQL-tietokantajärjestelmä muodostuu siis tietokannoista, jotka taas sisältävät tauluja. Taulut muodostuvat edelleen kentistä ja riveistä. Tässä työssä käytetään tietokantaa, joka on nimeltään ”srea_data” ja siellä sijaitsee taulukko nimeltään ”pvs300”. Kentät olen eritellyt taulukossa 1 sivulla 21. (Ohjelmointiputka, hakupäivä 19.6.2014.)

Ehkä eniten aikaa kului juuri tietokantakyselyjen muodostamiseen ja niiden saamiseksi sellaiseksi, että sain haluttu tiedon juuri tietyltä aikaväliltä. Alla olen eritellyt ja selittänyt käyttämäni MySQL-kyselyt. SELECT määrittää mitä kenttiä haetaan, FROM määrittää haettava tietokannan taulukon ja WHERE-osuus määrittää haulle ehdot.

```
SELECT date,time,`Power [kW]` FROM pvs300 WHERE `date` = (SELECT MAX(`date`) - INTERVAL 1 DAY FROM pvs300
```

Tämä kysely hakee päivämäärän, kellonajan ja tehontuoton eilisen päivän ajalta. Se on käytössä lähdekoodeissa 1 ja 2, eli päiväkohtaisen tehonseurannan käyrä ja tehontuoton pylväsdiagrammi.

```
SELECT WEEK(date),date,SUM(`Power [kW]`) FROM pvs300 WHERE WEEK(date,1) = WEEK(NOW() - INTERVAL 0 DAY) AND WEEKDAY(date) = '$x'
```

Tämä kysely hakee päivämäärän ja summaa haetut tehot yhteen päiväntuotoksi. Ajanväli on edellinen täysi viikko ja \$x muuttuja on viikonpäivien lukumäärä. Kysely on käytössä lähdekoodissa 3, eli tehontuoton esitys viikonajalta pylväsdiagrammi muodossa.

```
SELECT date,MONTHNAME(date) FROM pvs300 WHERE MONTH(date) = MONTH(NOW() - INTERVAL 1 Month
```

```
SELECT SUM(`Power [kW]`) FROM pvs300 WHERE MONTH(date) = MONTH(NOW() - INTERVAL 1 Month) AND DAY(date) = '$x'
```

Edelliset kaksi kyselyä hakevat päivämäärän ja kuukaudennimen sekä summaavat haetut tehot yhteen päiväntuotoksi. Ajanvälinä on kuukausi ja ”INTERVAL x Month” määrittää kuinka monta kuukautta mennään taaksepäin. Muuttuja \$x määrittää kuukaudenpäivien lukumäärän. Kyselyt ovat käytössä lähdekoodeissa 4, 5, 6 ja 7. Ne kuvaavat pylväsmuotoisina tehonpäiväntuotot nykyisen ja kolmen edellisen kuukauden ajalta.

```
SELECT YEAR(CURDATE()),SUM(`Power [kW]`) FROM pvs300 WHERE YEAR(date) = YEAR(CURDATE()) AND MONTH(date) = '$x'
```

Tämä kysely hakee nykyisen vuoden tiedon ja summaa haetut tehot yhteen kuukausi-tuotoksi. Ajanjakso on kuluva vuosi ja \$x muuttuja on vuodessa olevien kuukausien määrä. Kysely on käytössä lähdekoodissa 8 ja se kuvaa pylväsmuodossa tehonkuukausi-tuotot kuluvan vuoden ajalta.

```
SELECT date,`Power peak of day [kW]`,`Power peak time of day (Unix epoch) [s]`,`Power peak ever [kW]`,`Power peak time ever (Unix epoch) [s]`,`Total energy [kWh]` FROM pvs300 WHERE `date` = (SELECT MAX(`date`) - INTERVAL 1 DAY FROM pvs300)
```

Tämä kysely hakee päivämäärän, päivän korkeimman tehopiikin ja sen ajankohdan, koko tietokannan korkeimman tehopiikin ja sen ajankohdan sekä aurinkopaneeleilla tuotetun kokonaisenergian määrän. Tietoja käytetään lähdekoodin 10 sivulla tilastotietojen esittä-miseen.

```
SELECT date,YEAR(CURDATE()),SUM(`Power [kW]`),MONTH(CURDATE()) FROM pvs300 WHERE YEAR(date) = YEAR(CURDATE()) AND MONTH(date) = '$x'
```

```
SELECT date,SUM(`Power [kW]`) FROM pvs300 WHERE YEAR(date) = YEAR(CURDATE()) AND DAYOFYEAR(date) = '$x'
```

Ensimmäinen kysely hakee päivämäärän, nykyisen vuoden tiedon, summaa haetun kuu-kauden tehot yhteen ja hakee kuluvan kuukauden numerotiedon. Toinen kysely hakee päivämäärän ja summaa haetun päivä tehot yhteen. Muuttuja \$x on vuoden päivien luku-määrä, eli taulukkoon tallentuu tehontuotto vuoden joka päivä. Tietoja käytetään läh-dekoodin 11 sivulla tilastotietojen esittämiseen.

```
SELECT SUM(`Power [kW]`) FROM pvs300 WHERE YEAR(date) = YEAR(date('2014-01-01') + INTERVAL '$x' YEAR)
```

Tämä kysely hakee ja summaa vuoden ajalta tehot yhteen, alkaen vuodesta 2014. Muut-tuja \$x määrittelee sen, moneltako vuodelta tiedot haetaan. Kyselyä käytetään lähdekoo-dissa 9.

4.7 pChart kirjastot

pChart on PHP kirjasto, jonka avulla voi luoda graafisia diagrammeja tai kuvaajia suoraan nettiserverille. Nämä kuvaajat voi näyttää sellaisinaan selaimissa tai liittää esimerkiksi pdf-tiedostoihin. Käyttämäni kirjaston versio on pChart2.1.4 ja se on ei-kaupalliseen käyttöön ilmainen. pChart-kirjastot ovat maailmanlaajuisessa käytössä opiskelu ja tutkimus käytössä. Internetistä ja kirjaston omilta sivuilta löytyy runsaasti esimerkkejä erilaista pChart-kaavioista. Valitsin itse yhden esimerkkikoodin pohjaksi jota oli sitten helppo muokata omiin tarpeisiin sopiviksi, kunhan ensin oli perehtynyt asiaan. (pChart.net www-sivu, hakupäivä 24.6.2014).

Käytin kolmea eri luokkaa esitysten teossa, ne ovat pData, pDraw ja pImage. pData luokan avulla voi kustomoida piirrettävän sarjan tiedot ja akselien tiedot. Kaksi muuta luokkaa liittyy itse kuvaajan piirtämiseen. Alla olen luetellut pChart -kirjastosta käyttämäni funktiot. (pChart.net www-sivu, hakupäivä 24.6.2014).

Taulukko 2. Kuvaajissa käytetyt pChart kirjaston funktiot.

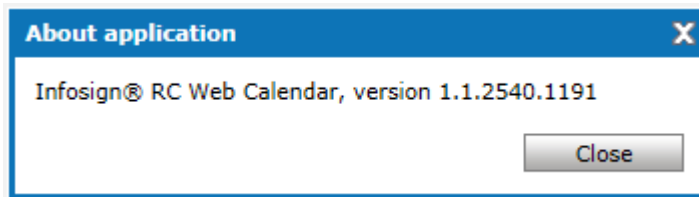
addpoints	- rakentaa piirrettävän sarjan annetuista arvoista
setAxisName	- määrittää Y-akselin nimen
setAxisUnit	- määrittää Y-akselin yksikön
setAxisDisplay	- määrittää missä muodossa arvot esitetään
setSerieDescription	- funktiota käytetään yhden sarjan kuvaukseen
setAbscissa	- funktion avulla yhdistetään sarjan nimi piirrettävään kaavioon
setPalette	- funktion avulla muokataan yhden sarjan väriasetuksia
drawFromPNG	- funktiota käytin taustakuvan asettamiseen
drawRectangle	- funktio piirtää nelikulmion
setFontProperties	- funktio määrittää käytettävän fontin asetukset
drawText	- funktio lisää tekstiosan kaavioon ja siinä myös määritellään sen asetukset
setGraphArea	- funktio määrittelee piirrettävän alueen fyysiset mitat ja sijainnin
drawFilledRectangle	- funktio piirtää täytetyn nelikulmion kaavion taustalle
drawScale	- funktio määrittää kuvaajan arvojen skaalauksen ja piirtää ristikon taustalle
setShadow	- funktio määrittää varjostuksen piirrettävään kohteeseen
drawBarChart	- funktio määrittää kuvaajan muodon pylväsdigrammiksi
drawSplineChart	- funktio määrittää kuvaajan muodon käyrämuotoon

Opinnäytetyössäni käytin kahta erilaista kuvaajatyyppeä. drawSplineChart -funktiota käytin kuvaamaan käyrämuotoista tehontuottoa vuorokauden ajalta. Muut kuvaajat toteutin käyttämällä drawBarChart -funktiota, joilla sain aikaan havainnollisen pylväs esityksen. Muut käytetyt funktiot ovat lähinnä esityksen ulkoasuun vaikuttavia tekijöitä.

5 INFOJÄRJESTELMÄ

5.1 InfoSign

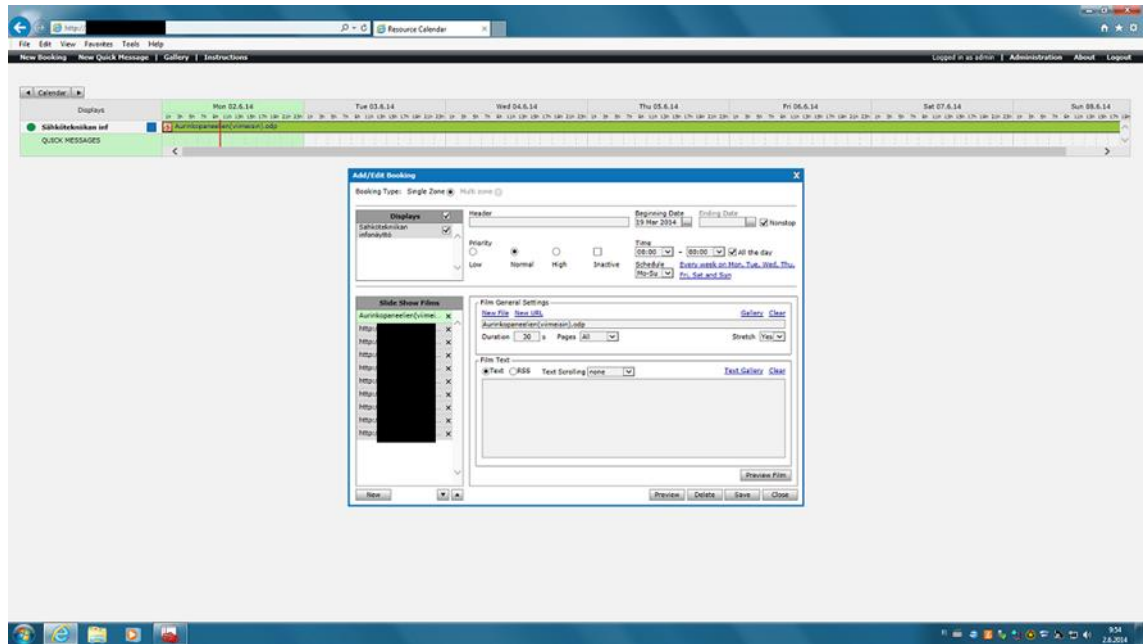
InfoSign-järjestelmän avulla voi ajastaa monipuolista sisältöä näytöille suoraan käytettävältä tietokoneelta tai internetistä. Sisältö voi olla kuvia, tekstitiedostoja, videoita, http-sivuja tai taulukoita. Näytöille voi antaa halutessaan eri oletustaustoja käyttäjien tai sijainnin mukaan. Näytön voi jakaa useaan eri itsenäisesti päivittyvään osaan. Lisäksi halutessaan voi ajastaa samaa tai eri sisältöä jokaiselle käytettävälle näytölle. Näitä ajastuksia voi tehdä useita samanaikaisesti tai eri prioriteetilla. Kaikki toiminnot, kuten käyttäjäoikeuksien hallinta, näyttöjen status, esikatselut ja ajastukset näkyvät selkeästi samalla kalenterinäkymällä. InfoSign-järjestelmällä on selkeä selainpohjainen hallinta ja niin sanottu drag-and-drop -ajastus. Alla olevasta kuvasta näkyy ohjelmiston versio tiedot. (InfoSign 2012, InfoSign infonäyttöjärjestelmä.pdf.)



Kuva 12. Ohjelman versio.

5.2 Käyttöohjeet

Opinnäytetyön yksi osa on laatia käyttöohje InfoSign-ohjelman käyttäjille samalla kun itse siihen tutustun. Kuvakaappaukset ovat koululla käytössä olevalta tietokoneelta, jolla ohjelmistoa ja tehonseuranta esitystä pyöritetään. IP-osoitteet ja muut vastaavat tiedot olen peittänyt mustilla palkeilla kuvista.



Kuva 13. Infojärjestelmän kalenterinäkymä.

InfoSign-järjestelmään kirjaudutaan selaimen kautta käyttämällä omia tunnuksia ja salasanoja. Lisäksi kirjautuessaan voi halutessaan valita käytettävän kielen. Aloitusruudussa näkyvät kalenterinäkymä, näyttöjen tiedot, yläosassa systeemikäskyt ja videomuotoiset ohjeet. Ajastuksen tai esitykselle voi tehdä kalenterille painamalla näytön nimeä tai vetämällä kalenterista ajastusalue, jolloin aukeaa ajastusikkuna. Ajastuksia voi muuttaa drag-and-drop -pohjalta tai painamalla vihreänä näkyvää ajastuspalkkia kalenterinäkymässä. Mustan yläpalkin New Booking -kohdasta voi myös aloittaa uuden ajastuksen. Yllä olevasta kuvasta 13 näkyy aloitusruudun kalenterinäkymä ja avattu ajastusikkuna. (InfoSign 2012, InfoSign infonäyttöjärjestelmä.pdf.)

Add/Edit Booking

Booking Type: ☒ Single Zone ☐ Multi zone

Displays ☒

- ☒ Sähkötekniikan infonäyttö

Header

Beginning Date 19 Mar 2014 **Ending Date** ☒ Nonstop

Priority
☐ Low ☒ Normal ☐ High ☐ Inactive

Time 00:00 - 00:00 ☒ All the day

Schedule Mo-Su [Every week on Mon, Tue, Wed, Thu, Fri, Sat and Sun](#)

Slide Show Films

- Aurinkopaneelien(viimeisin) x
- http://... val... x
- http://... val... x
- http://... val... x
- http://... val... x
- http://... val... x
- http://... val... x
- http://... val... x
- http://... val... x

Film General Settings
[New File](#) [New URL](#) [Gallery](#) [Clear](#)

Duration 30 s **Pages** All **Stretch** Yes

Film Text
☒ Text ☐ RSS **Text Scrolling** none [Text Gallery](#) [Clear](#)

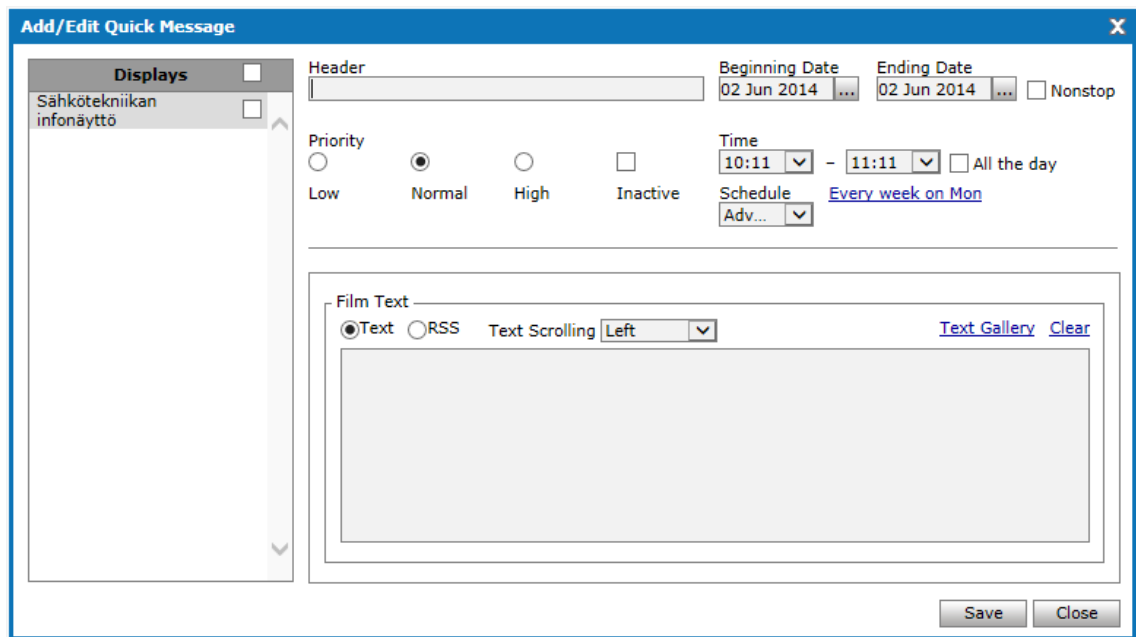
Kuva 14. Ajastusikkuna.

Ylläolevan kuvan ajastusikkunasta, Displays-kohdasta, voi valita, mille näytöille esitys näytetään. Header-kohta on esityksessä näkyvä yläotsikko ja sen alla on esityksen prioriteettitasen määrittäminen. Ajastuksille voi määrittää kolme eri prioriteettia: matala, normaali ja korkea. Näillä määrätään, mikä ajastus näytetään ensin. Saman prioriteetin omaavat ajastukset näkyvät samanaikaisesti. Oikealla puolella ylhäällä on esityksen ajastus määrittäykset. Näitä ovat esityksen aloitusajat ja päivämäärät sekä lopetusajat ja päivämäärät. Ajastuksia voi määrittää myös jollekin päivälle tai ajanjaksolle viikosta sekä onko esitys koko ajan pyörivä.

Film General Settings -osiosta voi lisätä sisältöä esitykseen tiedosto tai internet-sivu muodossa. Sisältöä voi ladata suoraan esitykseen omalta tietokoneelta, internetistä tai galleriasta. Filmin tai sisällön esityksajan ja sen, mitä niistä näytetään, voi määrittää sen nimipalkin alta. Film Text -osiosta voi esitykseen lisätä tekstisisältöä, joko tekstikirjastosta tai kirjoittamalla sen suoraan sille varattuun ruutuun. Tässä voi käyttää myös valmiita rss feed -pohjia. Vasemmalla puolella on Slide Show Films -osio, josta näkyy mitä sisältöä tai filmejä esitykseen kuuluu. Alareunassa on uuden sisällön lisäys ja niiden liikutteluun

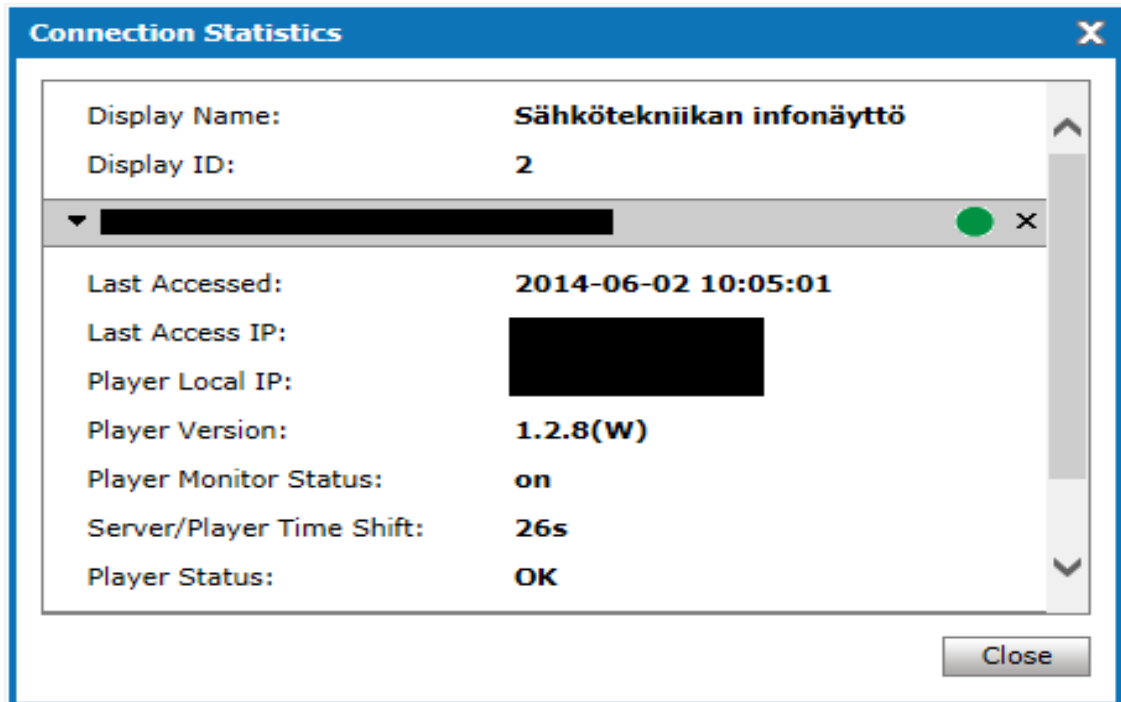
tarvittavat painikkeet. Oikealla alhaalla ovat esityksen esikatselu, poisto, talletus ja ajastusikkunan sulku painikkeet.

Gallery-ikkunassa on näkyvissä ja valittavissa kaikki InfoSign-ohjelmassa käytetyt nettisivut ja tiedostot. Sinne voi ladata ja tallentaa valmiiksi tiedostoja myöhempää käyttöä varten. Tallennetuista tiedostoista näkyy sen nimi, käyttäjän tunnus, muokkausaika ja tiedoston tyyppi. Tekstikirjasto on vastaavanlainen tallennustila tekstitiedostoille. Gallery-ikkunan voi myös avata mustasta yläpalkista.



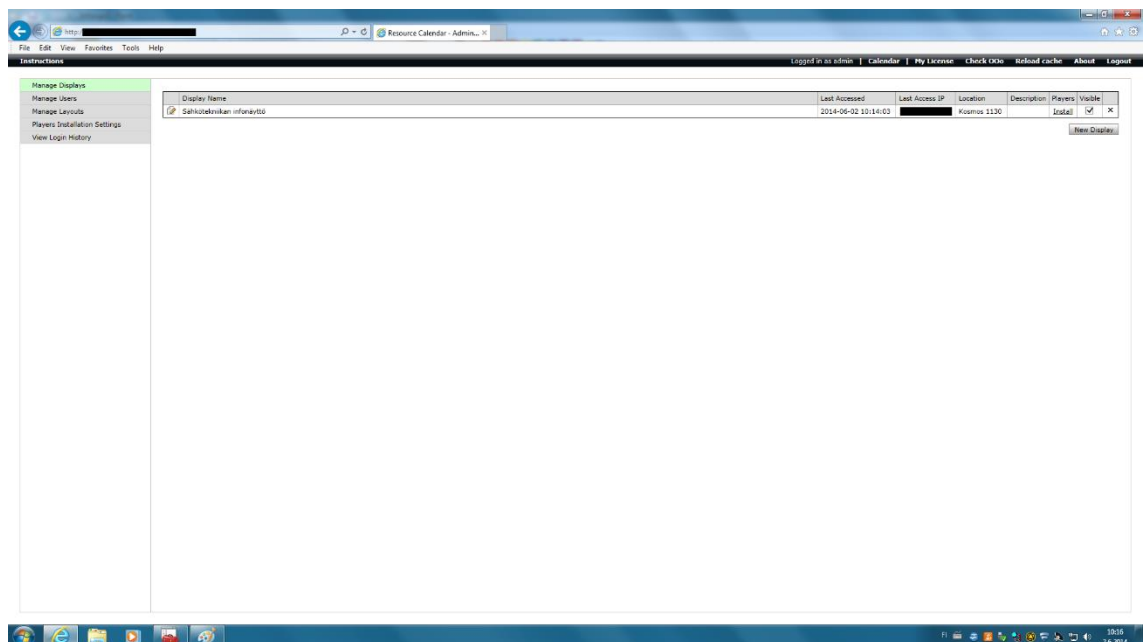
Kuva 15. Pikaviestin ikkuna.

Mustassa yläpalkissa olevasta New Quick Message -kohdasta voi avata yllä näkyvän ikkunan. Sen avulla voi lisätä helposti muuteltavan viestin tai tekstin näkymään esitykseen sen aikana. Tekstin voi kirjoittaa ikkunassa olevaan ruutuun tai sen voi poimia tekstikirjastosta. Tekstin tai viestin voi myös laittaa liikkumaan ruudulla eri suuntiin. Muut pikaviestin määrittelyt ovat samat kuin edellisessä ajastusikkunassa.



Kuva 16. Näytön statusikkuna.

Kalenterinäkymästä vasemmalta löytyy Displays-kohta. Siellä sijaitsee tiedot kaikista käytettävistä näytöistä. Vihreää ympyrää painalla avautuu Connection Statistics -ikkuna. Sininen neliö taas näyttää miten esitys pyörii käytettävässä näytössä. Quick Message -kohta taas avaa pikaviesti-ikkunan.



Kuva 17. Administration-ikkuna.

Mustassa yläpalkissa oikealla puolella on tieto kirjautuneesta käyttäjästä, ohjelman hallintapainike, tiedot ohjelmasta ja uloskirjautumispainike. Administration-ikkunasta voi hallita näyttöjä, tietoja käyttäjistä, ohjelman ulkoasua ja sen toimintaan vaikuttavia seikkoja sekä sieltä voi nähdä historian ohjelman käytöstä.

Manage Displays -kohta sisältää tiedot näyttöjen nimistä, viimeisistä yhteyksistä niihin, näytön IP-osoitteen, sijainti tiedon, kuvauksen ja onko kyseinen näyttö käytössä. Sieltä käsin voi myös lisätä uusia näyttöjä.

Manage User -kohta sisältää tietoja ohjelman käyttäjistä, kuten nimet, puhelinnumerot, email -osoitteet ja millaisia oikeuksia käyttäjällä on.

Manage Layouts ja Players Installation Settings -kohdat määrittelevät ulkoasu tekijöitä ja ohjelmaa pyörittävän sekä päivitettävän serverin IP-tiedot.

View Login History -kohta sisältää ohjelman käytön historia tiedot, kuten kirjautuneen käyttäjän tunnuksen, kirjautuneen tietokoneen IP-osoitteen, kirjautumisen ja uloskirjautumisen ajat. Sieltä on myös mahdollista tyhjentää historia, mikäli siihen on oikeudet.

6 AURINKOPANEELEIDEN TEHON SEURANTA

Tässä osuudessa selvitän työni tulokset ja mitä niihin liittyi. Tarkoitus on tutustua aurinkopaneeleiden sähköntuotantoon ja tehdä niistä graafinen esitys, joka näkyy tekniikan yksikön käytävällä inforuudussa ja koulun nettisivuilla. Näissä olisi tarkoitus myös saada näkymään tekniikan yksikön sääaseman tiedot. Lisäksi laadin infojärjestelmän käyttäjille käyttöohjeen ja selvityksen sen toiminnasta. Muita selvitettäviä asioita ovat infojärjestelmän serveriin ja tietojen varmuuskopiointiin liittyvät seikat.

6.1 Infojärjestelmän serveri

Yksi asia, mikä piti selvittää, oli mihin infojärjestelmän serverin voisi sijoittaa ja kelle tunnuksat ja salasanat annetaan tietoon. Nykyinen serverin paikka on sähkövoimatekniikan laboratoriossa, jossa liikkuu paljon eri henkilöitä. Itse serveri on nykyisin siis alttiina ulkoisille uhille ja sille parempi paikka olisi lukittu verkko ja sähköjakamo, jonka huone-tunnus on 1156. Toinen hyvä syy siirtää serveri on sen jatkuva käynnissä olon tarve. Kun nykyisestä sijaintipaikasta katkaistaan virrat, niin serverissä sijaitsevat tietokannan tiedot eivät päivitty. Tämä näkyy esityksessä puuttuvina tai vajaina pylväinä. Tunnuksat ja salasanat voisi antaa IT-osaston haltuun.

MySQL-tietokannan tiedot voi manuaalisesti täydentää hakemalla ja lataamalla ne ABB srea:n ohjelmistosta csv-tiedostona. Ladatusta csv-tiedostosta voi Excel-ohjelman avulla etsiä ja leikata puuttuvat tiedot omaan tiedostoon. Paikkaus csv-tiedosto sijoitetaan serverikoneella olevaan 'ABB SREA Logi Upload' -kansioon. Sieltä tiedosto poimitaan automaattisesti määrätyn ajan kuluttua ja sen tiedot päivittyvät tietokantaan.

6.2 Varmuuskopiointi

Toinen selvitettävä asia on, mitkä tiedostot ja kansiot pitäisi varmuuskopioida. Varmuuskopiotavia kansioita serverillä olisivat ainakin 'ABB srea-50', 'Xampp' ja 'www'. Eri-tyisesti 'ABB srea-50' -kansiossa sijaitseva 'ABB SREA Logi' -kansio on tärkeä, sillä siinä sijaitsee tietokannan csv-tiedostot. 'Xampp' -kansiossa sijaitsee itse MySQL tietokanta ohjelmisto. 'www' -kansiossa ovat tehdyt php-sivujen koodit ja niihin liittyvät muut tiedostot. Nämä kansiot voisi ensin tallentaa erillisille massamuisteille tai tikuille. Mutta

verkonvälityksellä ajoittain tapahtuva varmuuskopiointi olisi parempi ratkaisu. Nykyisin kaikki tiedot tai tiedostot ovat vain yhdellä käytössä olevalla serveri koneella.

6.3 PHP-sivut

Tein yhdeksän erilaista php-sivua, jotka kuvaavat graafisesti aurinkopaneelien tehon ja energian tuottoa eri ajanjaksoina. Lisäksi kaksi sivua esittää erilaisia tilastotietoja niiden toiminnasta teksti ja numerotietoina. Esityksen kieleksi valitsin englannin kielen. Kuvaajin taustana dokumentin kuvissa on pelkkä harmaa tausta, mutta sen voi vaihtaa helposti toiseen taustakuvaan vaihtamalla vain tiedostonnimen tai -polun koodiin. Taustakuvan koko on oltava tarkalleen 1920 * 1080.

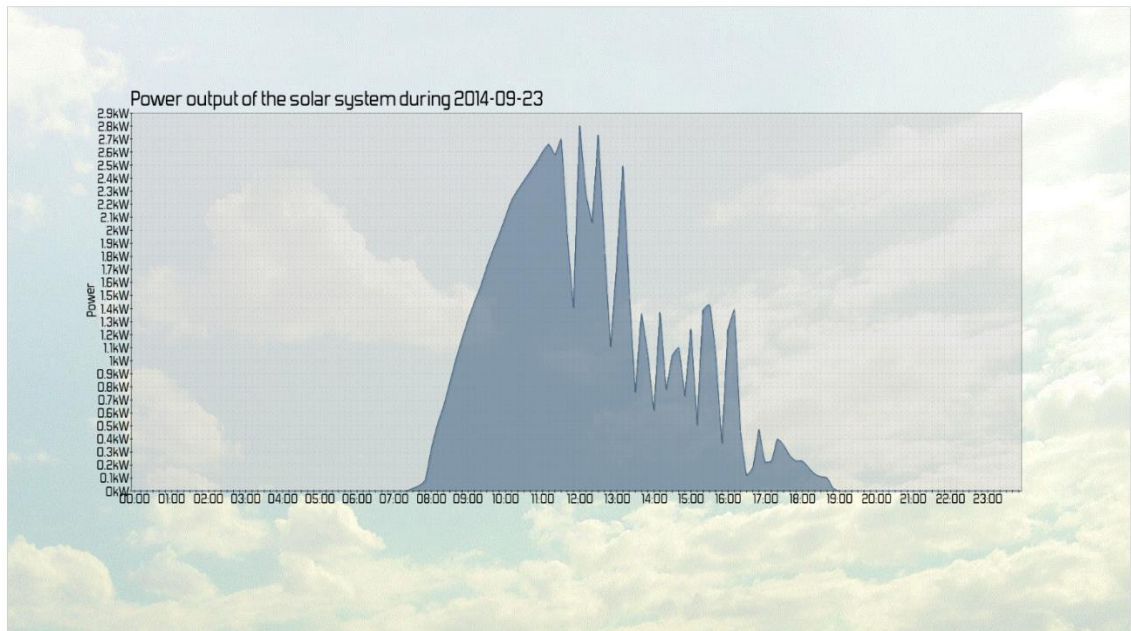
Sivujen rakenne on pääasiassa seuraava: ensin kirjastojen sisällyttäminen, \$Mydata muuttujan luominen ja siihen pData objektin sijoittaminen, MySQL-tietokantaan yhdistäminen ja siihen tehtävä hakukysely, kyselyistä muodostetaan merkkijonomuuttujat ja niitä muokataan tarpeen mukaan, muuttujan tiedot sijoitetaan \$Mydata muuttujaan, sitten seuraa kuvaajan ulkonäköön liittyviä määrittelyjä ja lopulta itse kuvaaja muodostetaan. Teke- mieni php-sivujen lähdekoodit löytyvät liitteinä dokumentin lopusta.

6.3.1 Päiväkohtainen tehon seuranta

Tekemäni esityksen ensimmäiset kaksi php-sivua näyttää aurinkojärjestelmän tehon tuoton graafisesti käyrämuodossa edellisen ja kuluvan vuorokauden ajalta. Näytteitä tietokannassa on kymmenen minuutin välein, ja niistä muodostetaan merkkijono muuttuja \$Power. Muuttujan ja pChart kirjaston funktion drawSplineChart() avulla muodostetaan näytteistä käyrämuotoinen kuvaaja. \$Date muuttujaan haen viimeisimmän näytteen päivämäärän ja se näkyy kuvaajan otsikossa. Y-akselille annoin nimen Power ja se on skaalautuva näytteiden tehon suuruuden mukaan. Y-akselin yksiköksi on määritetty kW, eli lyhennys kilowatista. X-akselille on annettu kellonajat vuorokauden alusta loppuun.

Koodissa on ehtolause, joka korjaa \$Power muuttujan arvon nolnaan, jos sen arvo on alle sen. Muuten käyrän muoto ja sen tietosisältö vääristyy. Syynä negatiivisiin arvoihin voisi

olla järjestelmän oma kulutus. Eroa näissä kahdessa php-sivussa on vain MySQL-kyse-lyssä käytetty ajanjakson muuttuminen yhdellä vuorokaudella. Päiväkohtaisen tehon- tuoton lähdekoodin olen merkinnyt numero ykköseksi ja kuva php-sivusta on alla.



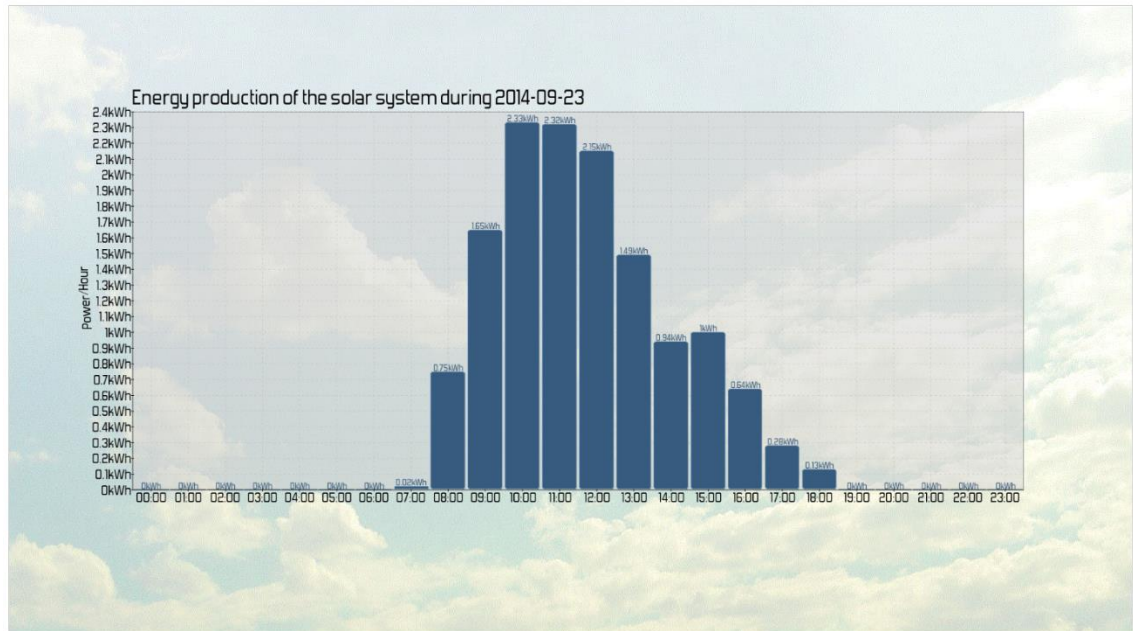
Kuva 18. Päiväkohtainen tehontuoton seuranta.

6.3.2 Päiväkohtainen energiantuotto

Tekemäni toinen päiväkohtainen kuvaaja näyttää aurinkopaneeleiden energiantuoton vuorokauden ajalta pylväsmuodossa. Tarvittavat tiedot haetaan tietokannasta 24:llä eri kyselyllä ja yksittäisellä kyselyllä tiedot summataan yhteen yhden tunnin energian tuotoksi. Yhden tunnin energian tuoton tiedon tallettamiseen käytän \$PowerHour-nimistä muuttujaa ja niitä on taulukoitu 24 kappaletta, eli yksi joka vuorokauden tunnille. Tämän summatun muuttujan taulukon arvoja jouduin vielä korjaamaan erillisellä silmukalla, jossa jaoin muuttujan arvon kuudella ja korjasin negatiivisen arvon nolllaksi sekä pyöristin samalla arvon kahden desimaalin tarkkuuteen.

\$PowerHour muuttujan ja pChart-kirjaston funktion drawBarChart() avulla muodostuu yllä olevan näköinen graafinen pylväskuvaaja. Muuttuja \$Date sisältää viimeisimmän näytteen päivämäärän ja sitä käytän otsikon päivämäärän esittämiseen. Y-akselilla on skaalautuvasti energiantuoton asteikko ja yksikkönä on kWh, eli kilowattitunti. X-akselilla on vuorokauden tunnit alusta loppuun. Pylväiden päällä on energian tuntituottotieto

numeromuodossa. Päiväkohtaisen energiantuoton lähdekoodin olen merkinnyt numero kakkoseksi ja kuva php-sivusta on alla.



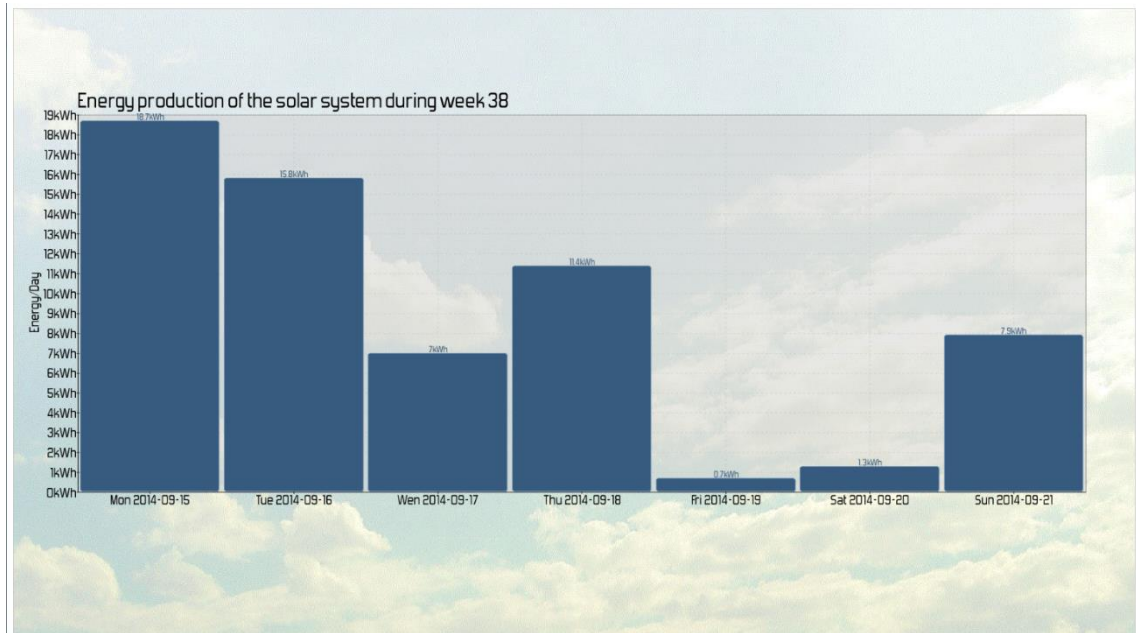
Kuva 19. Päiväkohtainen energiantuoton seuranta.

6.3.3 Viikkokohtainen energiantuotto

Kolmas tekemäni kuvaaja näyttää energiantuoton edellisen täyden viikon jokaiselta päivältä pylväsmuodossa. Kuvaajaan tarvittavat tiedot haetaan tietokannasta seitsemällä eri kyselyllä ja yksittäisessä kyselyssä summataan yhteen vuorokauden ajalta tehontuotot sekä haetaan päivämäärätieto että viikon numerotieto. Vuorokauden energiantuotot tallennetaan taulukkoon \$PowerDay[].

Viikkokuvaaja esitetään kyseisen viikon numeron mukaan, alkaen maanantaista ja lopuen sunnuntaihin. Viikkojen numerointi määritellään WEEK(date) haku kyselyllä tietokannasta ja tieto tallennetaan muuttujaan \$Week. Viikon numerotietoa käytetään lähinnä vain oikean tietokantakyselyn muodostamiseen sekä kuvaajan otsikossa. Tietokannasta haetaan myös päivämäärätieto jokaiselle vuorokaudelle taulukkoon \$Date[]. Päivämäärät näkyy myös X-akselin viikonpäivien nimien vieressä ja Y-akselille on sijoitettu skaalautuvat energiantuoton yksiköt.

Kuten edellisessäkin kuvaajassa, täytyy vuorokauden tuotot jakaa kuudella ja tehdä pyöristykset, jotta tulokset ovat oikeat. Tiedot energiantuotonmääristä vuorokaudenajalta näkyy numerotietona myös pylvään yläpuolella. Viikkokohtaisen energiantuoton lähdekoodin olen numeroinut kolmoseksi ja kuva php-sivusta on alla.



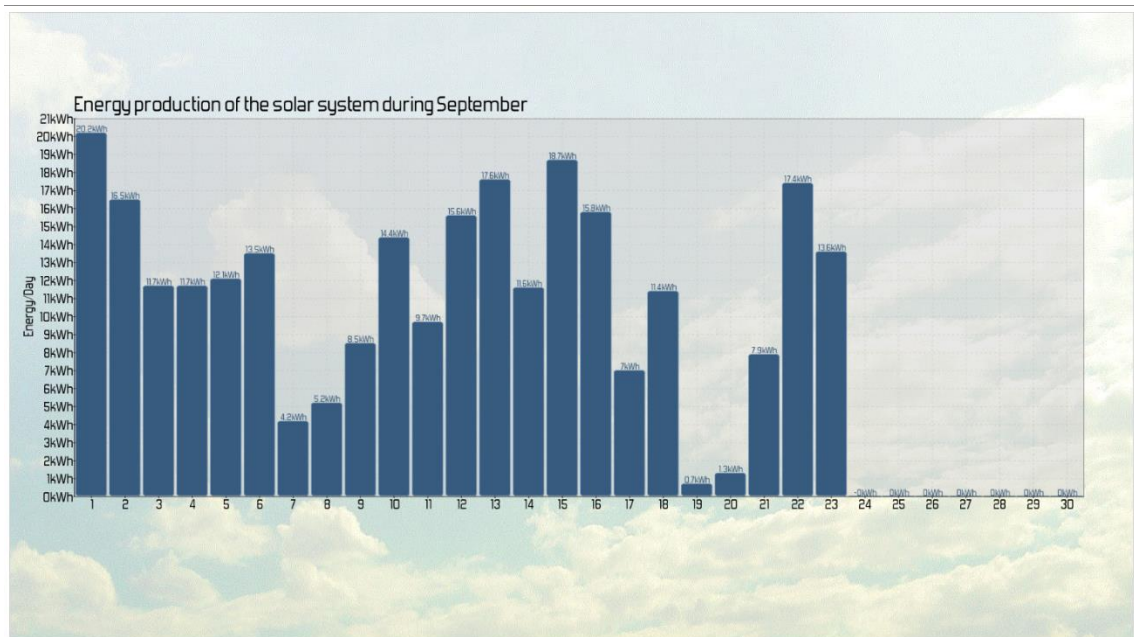
Kuva 20. Energiantuotto edellisen viikon ajalta.

6.3.4 Kuukausittainen energiantuotto

Seuraavat neljä tekemääni pylväskuvaajaa esittävät aurinkojärjestelmän vuorokauden energiantuoton koko kuukauden ajalta. Ensimmäinen näistä kuvaa kuluvaa kuukautta ja kolme muuta edellisiä kuukausia.

Sivut ovat rakenteeltaan samalaisia, vain haku kyselyssä on hieman eroa. Tietokantakyselyjä on kaksi erilaista. Ensimmäisessä kyselyssä haetaan päivämäärä ja kuukauden nimi. Päivämäärän mukaan määritellään kuukauden vuorokausien määrä muuttujaan \$daymax. Kuukauden nimi sijoitetaan muuttujaan \$Month ja sitä käytetään kuvaajan otsikossa. Toinen tietokantakysely tekee kuukauden vuorokausien mukaisen määrän summattuja energiantuottoja ja ne sijoitetaan taulukkoon nimeltä \$PowerDay[]. Sen jälkeen taulukon tulokset jaetaan kuudella ja pyöristetään yhden desimaalin tarkkuuteen.

Y-akselin asteikko on skaalautuva ja se näyttää energiantuottoa vuorokauden ajalta. X-akselilla on silmukalla toteutettu kuukauden vuorokausien numerointi. Pylvään yläpuolella näkyy vuorokauden energiantuotto numerotietona. Kuluvan kuukauden lähdekoodin olen merkinnyt numeroksi 4. Siitä edellisen kuukauden olen merkinnyt numerolla 5 ja seuraavat lähdekoodit numeroilla 6 ja 7. Kuva php-sivujen ulkoasusta on alla.



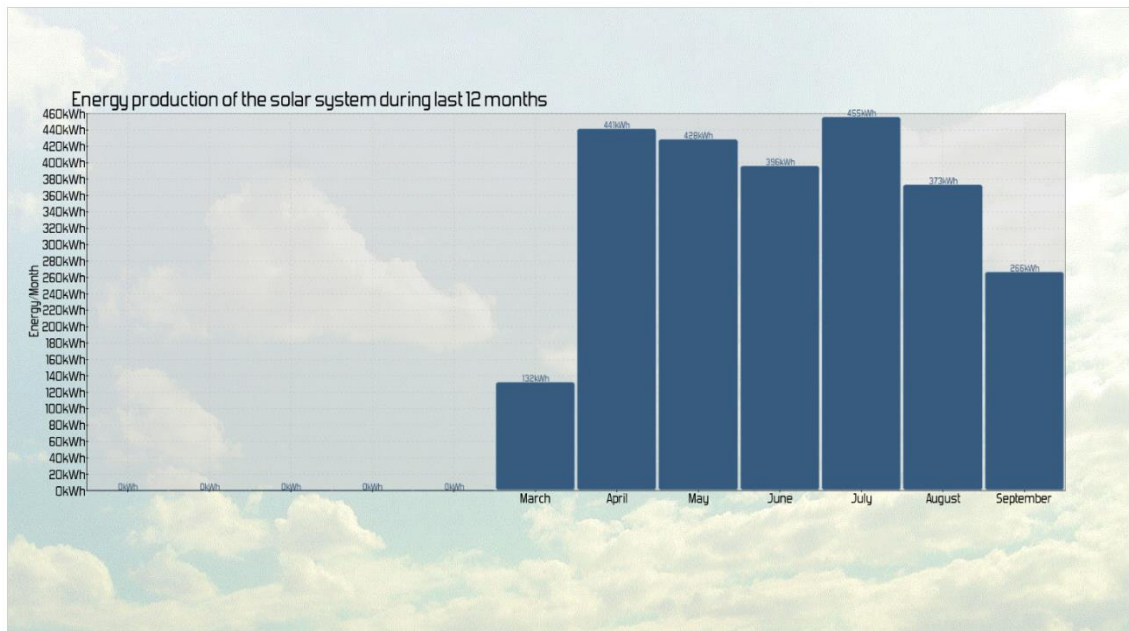
Kuva 21. Kuukausikohtainen energiantuotto.

6.3.5 12 kuukauden energian tuotot

Toiseksi viimeinen pylväskuvaaja näyttää aurinkojärjestelmän energiantuoton jokaiselta kuukaudelta vuoden ajalta. Kuvaajaan haetaan tiedot kahdellatoista eri kyselyllä ja yksittäisessä kyselyssä summataan kuukauden ajalta tehontuotot yhteen ja haetaan samalla kuukausien nimet taulukkoon \$Month[].

Summattu energiantuotto tallennetaan taulukkoon \$PowerMonth[]. Taulukon arvot jaetaan kuudella ja ne pyöristetään kokonaisluvuiksi, minkä jälkeen niistä muodostetaan drawBarChart-funktion avulla pylväskuvaaja.

Y-akselilla on skaalautuva kuukauden energiantuotto asteikko ja X-akselilla kuukauden nimet. Pylvään yläpuolella näkyy kuukauden energiantuotto numerotietona. Kuluvan vuoden energiantuoton sivun lähdekoodin olen merkinnyt numerolla 8 ja kuva php-sivun ulkoasusta löytyy seuraavalta sivulta.



Kuva 22. 12 kuukauden energiantuotot.

6.3.6 Vuosien energiantuotot

Viimeinen tekemäni php-sivu näyttää energian tuotot vuosilta 2014-2020. Tiedot haetaan jokaiselle vuodelle omalla kyselyllä, ja ne summataan yhteen vuosien tuotoiksi taulukoon `$PowerYear[]`. Taulukon arvot jaetaan kuudella ja ne pyöristetään kokonaisluvuiksi, minkä jälkeen niistä muodostetaan `drawBarChart` -funktion avulla pylväskuvaajat.

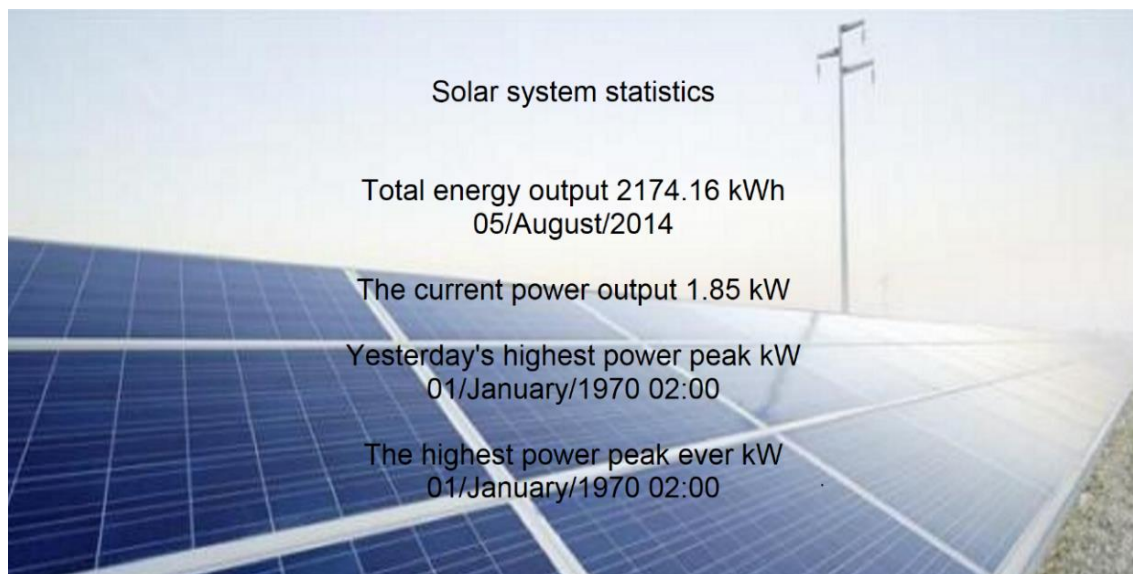
Y-akselilla on skaalautuva vuoden energiantuotto ja X-akselilla on vuosiluvut. Pylvään päällä näkyy vuoden energiantuotto numerotietona. Vuosien energiantuoton olen merkinnyt numerolla 9 ja kuva vuosien energiantuotosta php-sivusta löytyy seuraavalta sivulta.



Kuva 23. Nykyisen ja tulevien vuosien energian tuotot.

6.3.7 Tilastosivut

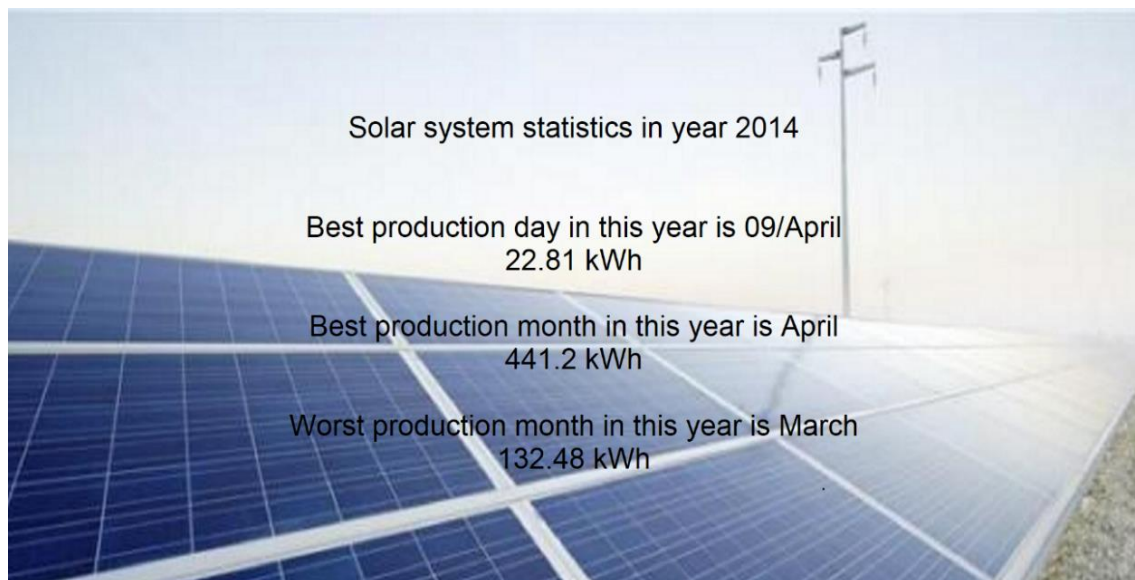
Tekemäni kaksi muuta php-sivua näyttää teksti ja numerotietoina erilaisia tilastotietoja aurinkojärjestelmän toiminnasta. Tiedot haetaan joko suoraan tietokannasta tai ne lasketaan saatavissa olevista tiedoista. Tilastosivuille on asetettu taustakuva, jota voi vaihtaa muuttamalla tiedostonnimen tai polun. Kuva Statistic php-sivusta on alla.



Kuva 24. Statistic-sivu.

Ensimmäisellä Statistic-sivulla näkyy järjestelmän kokonaisenergiantuotto kilowattitunteina sen elinkaaren aikana ja katseluhetken päivämäärä. Sen alla näkyy aurinkopaneelin katseluhetken tehontuotto kilowatteina. Seuraavat tiedot ovat eilisen päivän korkein tehoipikki ja sen ajankohtatieto sekä kaikkien aikojen korkein tehoipikki ja sen ajankohtatieto.

Sivulle olen tehnyt kaksi eri tietokantakyselyä. Ensimmäisessä kyselyssä haetaan tiedot tehoipikkeistä ja niiden ajankohdista. Toinen kysely hakee tiedot sen hetken tehontuotosta, päivämäärän ja tiedon kokonaisenergiantuotosta. Sivu sisältää runsaasti erilaisia muuttujia, joihin on sijoitettu erilaisia ajankohtia, tehon- ja energiantuottotietoja. Lähdekoodin olen merkinnyt numerolla 10.



Kuva 25. Statistic2 -sivu.

Toinen Statistic2 -sivu näyttää tiedot parhaasta energiantuoton päivästä ja kuukaudesta sekä tiedon huonoimmasta energiantuoton kuukaudesta. Lisäksi näistä on näkyvillä päivämäärä tai kuukaudennimi. Tämä sivu sisältää kaksi eri tietokantakyselyä ja runsaasti eri muuttujien keskenään vertailua taulukoissa, jotta niistä löytyisi korkeimmat ja huonoimmat arvot. Ensimmäisellä kyselyllä haetaan kuukauden- ja vuodentiedon sekä hakee päivämäärät ja summaa yhteen kuukausitehot taulukkoihin 12 eri muuttujaan. Toinen kysely hakee päivämäärät ja summaa yhteen vuorokausitehot taulukkoihin 366 eri muuttujaan.

Vertailut on toteutettu for-silmukalla ja ehtolauseella, joissa käydään läpi kuukausien energiantuotot ja vertaillaan niitä keskenään. Kun niistä löytyy huonoin ja paras kuukausi, tiedot energiantuotosta ja sen ajankohdasta sijoitetaan erillisiin muuttujiin. Paras päivä-tuotto on toteutettu samalla tavalla. Energiantuoton taulukko arvoja pitää käsitellä samalla tapaa kuin edellisissäkin sivuissa, jotta tulokset olisivat oikeat. Tämän sivun lähdekoodi on numero 11.

6.3.8 Sääasema

Sääasema on Schneider Electric Industries SAS:n valmistama yhdistelmäyksikkö, jota voidaan käyttää tuulen, sateen, kirkkauden ja lämpötilan tunnistimena. Itse laite on malliltaan KNX perussääasema. Itse tunnistinarvot ja mittausarvot käsitellään laitteessa. Sääasemaa käytetään KNX väylän kanssa, johon mittausarvot lähetetään suoraan. Mittausarvoja voi tarkastella IP-reitittimen välittämänä koululle asennetun ETS4 ohjelmiston avulla. Mittausarvoja ei saa suoraan KNX järjestelmästä erilliseen tietokantaan, vaan siihen tarkoitukseen on käytettävä omaa rajapintaa.

Taulukko 3. Sääaseman tekniset tiedot.

Mittausalue:	-20°C ... +55°C
Kirkkausala:	1 - 100 000 lux
Käyttöjännite:	230-240 V AC/50-60 Hz
Virrankulutus:	< 10 mA väyläjännitteelle
Verkkovirran kulutus:	10 W lämmityksen ollessa kytkettynä
Kotelointiluokka:	IP 44 normin EN 60529 mukaisesti
Toimintatapa:	RS tyyppi 1 normi EN 60730-1
Suojaluokka:	II oikein asennettuna
Tunnistuskulma:	150°
Mastokiinnitys:	Dm 60-80 mm (lisävaruste)

Jotta sääaseman tiedot saisi KNX-järjestelmästä MySQL-tietokantaan, on käytettävä siihen kehiteltyjä rajapintoja. Tähän tarkoitukseen on kehitelty laite- ja ohjelmistoratkaisuja. Olen etsinyt ensin kaupallisia rajapintoja ja BAB technologie nimisellä yrityksellä on useampi vaihtoehto tähän. Kyseinen yritys valmistaa eibPort version 3 ja Datalogger nimellä olevia laitteita, joilla voisi KNX-järjestelmästä siirtää tiedot tietokantaan. eibPort version 3 on yhdyskäytävä, johon on saatavissa DataWarehouse 2.0 -niminen lisäohjelma. Tätä ohjelmaa voi käyttää tiedon siirtoon tietokantaan. Tutkin myös paria ilmaisohjelmaa, joilla voisi sääasemantiedot myös saada käyttöön. Calibri on java pohjainen

ilmaisohjelma, jolla voi siirtää tiedot tietokantaan. Toinen löytämäni ohjelma on nimeltään KNX Logger, jolla voi sääaseman tiedot sisällyttää suoraan csv-tiedostoon.



Kuva 26. eibPort version 3 ja BAB DataLogger.(BAB technologie www-sivu)

BAB technologiien valmistama ja KNX-järjestelmässä toimiva eibPort version 3 yhdyssäilyvä on laite, jolla yhdistetään ja hallitaan verkon välityksellä kodin viihdetoimintoja. Sen verkkokäyttöliittymän kautta on mahdollista siis hallita erilaisia kodintoimintoja ja siihen voi lisäksi asentaa DataWarehouse 2.0 niminen lisäohjelma, jolla voi KNX-dataa käsitellä ja tallentaa MySQL-tietokantaan. Tämä laite mahdollistaa kodintoimintojen etäkäytön verkossa PC:n tai mobiililaitteiden kautta sekä KNX-järjestelmän ilmoitukset ja viestit myös lähetetään niihin. Lisäksi laite mahdollistaa myös ETS ohjelmoinnin etätoimintana. DataWarehouse 2.0 ohjelmalla voi käsitellä, tutkia ja tallentaa tietokannassa tietoa rajattomasti. Se käyttää tiedonsiirtoon samaa csv-tiedostomuotoa kuin srea. (BAB Technologie www-sivu, hakupäivä 24.7.2014).

Toinen saman yrityksen valmistama laite on BAB DataLogger. Kyseisen laitteen käyttö tarkoitus on tallentaa KNX-telegrammeja ja se on suoraan yhdistettävä KNX-väylään tai se saa datan KNXnet/IP protokollan kautta verkosta. Kaiken datan voi tallentaa sen sisäiseen muistiin tai siirtää ulkoiseen MySQL-tietokantaan. Datan säilytys- ja siirto aikoja voi vapaasti muokata, samoin kuin sitäkin, minkä tyyppistä dataa tallennetaan. Talletetun datan voi myös siirtää verkossa ja USB-muistitikulla csv tai xls -tiedostoina. (BAB Technologie www-sivu, hakupäivä 29.7.2014).

Ilmaisohjelmista ensimmäisenä löysin javalla toteutetun, Calibri -nimisen avoimen lähdekoodin ohjelman, jolla voi lukea ja kirjoittaa telegrammeja EIB/KNX väylässä. Ohjelman tarkoitus on vaikuttaa koko väylä järjestelmää sillä tavalla, että KNX:n käyttämät

telegrammit voisi siirtää MySQL-tietokantaan. Sen toiminta perustuu Calimero kirjastoihin, jotka käsittelevät kommunikaatiota itse Calibrin ja EIB yhdyskäytävän välillä. Toimiakseen ohjelma tarvitsee KNXnet/IP reitittimen sekä serverin jossa tietokanta ja ohjelma sijaitsevat. Toinen ilmaisohjelma on nimeltä KNX Logger. Tämä ohjelma kerää dataa KNX väylästä myöhempää analysointia varten. Ohjelma tallentaa datan csv-tiedostona ja sitä voi helposti jatkossa käsitellä. Ohjelma ei suoraan tallenna dataa tietokantaan. (Sourceforge [www-sivut](#), hakupäivä 29.7.2014).

6.4 Työn arviointi

Työni lopputuloksen arvioisin onnistuneen ihan hyvin. PHP-sivujen toiminnassa löytyi muistin hallintaan liittyvä ongelma kesän jälkeen. Sivujen grafiikka katosi jonkin ajan kuluttua ja kaatui lopulta virheilmoitukseen. Sen ongelman selvitin ja tein tarvittavat korjaukset lähdekoodeihin. Lisäsin php-sivuilla muistin käytön määrää ja muutin tietokanta kyselyjen muotoa muistia säästävämmiksi. Muutin kuvaajien taustakuvan gif-muotoiseksi, mikä vähentää muistin tarvetta php-sivuilla. Lisäksi koodin lopussa tyhjennän käytetyt muuttujat ja taulukot. Sivut näyttäisivät toimivan nyt vakaammin. Sääaseman tiedot jäivät pois esityksestä, koska niiden siirtoon tietokantaan tarvitaan erillinen rajapinta. Itse sitä en voisi sitä jo ajankäytöllisistä syistä tehdä. Hyviä valmiita ja kaupallisia ratkaisuja toki löytyy. Muita suurempia ongelmia ei tullut vastaan.

7 POHDINTA

Tekemäni opinnäytetyö käsitteli aurinkopaneeleiden tehon ja energian tuoton seuranta graafisena esityksenä koulun infojärjestelmässä ja nettisivuilla. Toinen tehtävä oli infojärjestelmästä laatia käyttöohje. Lisäksi mietin infojärjestelmän serverin paikkaa sekä sen tunnusten ja salasanojen haltijaa. Omasta mielestä työn teko onnistui ihan hyvin, tekemäni php-sivut alkoivat toimia ja näkyä miten oli tarkoituskin. Opin sivuja tehdessä paljon php-ohjelmoinnista, tietokannoista ja muita aurinkojärjestelmiin liittyviä asioita. Edeltävistä opinnoista ja käydyistä kursseista oli varsin paljon apua, kun jo tiesi joitain käsiteltävistä asioista. Oppimista tapahtui myös infojärjestelmään tutustuessa ja sen käyttöohjetta laatiessa.

Näkyvin osa opinnäytetyöstä oli saada tekemäni graafinen esitys pyörimään näytössä. Esityksessä oli kymmenkunta erilaista php-sivua, jotka näyttivät koulun aurinkojärjestelmän tehon ja energiatuottoa eri ajanjaksoina. Tiedot näiden sivujen toteutukseen hain tähän tarkoitukseen tehdystä tietokannasta. Työni tekee aurinkopaneeleiden seurannan helpommaksi, ja kuka tahansa voi nyt seurata niiden toimintaa joko koulun infojärjestelmästä tai nettisivuilta.

Pysyin tekemässäni aikataulussa ja törmäsin vain yhteen isompaan ongelmaan. Ongelmana oli käytetyn tietokannan tiedoissa olevat aukot, joita tuli, kun käytetyn serverin virta oli katkennut. Kaikki tarvittavat tiedot kuitenkin löytyvät srea:n omasta muistista, josta ne pystyttiin siirtämään tietokantaan. Puuttuvat tiedot näkyivät pylväskuvaajissa vajaina tai puuttuvina pylväinä.

Mitä olisin tehnyt toisin, olisi parempi suunnitteluvaihe kun aloin tehdä php-sivuja. Jouduin tekemään muutaman sivun uudelleen, kun huomasin alkuperäisen idean tekevän virheellisiä pylväskuvaajia. Mutta virheistähän sitä juuri oppii. Sääaseman tiedot olisi pitänyt saada myös mukaan esitykseen, mutta se vaatisi oman rajapinnan KNX väylästä tiedon ulos saamiseksi. Tutkin kuitenkin, mitä eri mahdollisuuksia tähän tarkoitukseen olisi. Kaikin puolin olen varsin tyytyväinen saavuttamaani lopputulokseen, oppimiseen ja työtulokseen.

LÄHTEET

- ABB www-sivut 2014. Hakupäivä 1.7.2014.
<<http://www.abb.fi/industries/fi/9AAC166916.aspx?country=FI>>
- BAB Technologie www-sivut 2014. Hakupäivä 24.7.2014.
<<http://www.bab-tec.de/index.php/products.html>>
- Finnwind Oy 2013. Aurinkoenergiaopas.pdf. Hakupäivä 9.6.2014.
<<http://www.finnwind.fi/aurinko/Aurinkoenergiaopas-Finnwind.pdf>>
- Käytännön php-opas www-sivut 2014. Hakupäivä 1.6.2014.
<http://users.metropolia.fi/~juhavan/php/putka/php_8.php>
- Moodle www-sivut 2014. 3S2586_työelämäprojekti_1_k2014, aurinkopaneelijärjestelmä työ 11, ABB PVS300 string inverter manual.pdf. Hakupäivä 16.6.2014.
<<http://moodle.tokem.fi>>
- Moodle www-sivut 2014. 3S2586_työelämäprojekti_1_k2014, aurinkopaneelijärjestelmä työ 11, ABB SREA-50 remote monitor adapter.pdf. Hakupäivä 16.6.2014.
<<http://moodle.tokem.fi>>
- Moodle www-sivut 2014. 3S2586_työelämäprojekti_1_k2014, aurinkopaneelijärjestelmä työ 11, InfoSign speksit.pdf. Hakupäivä 26.6.2014.
<<http://moodle.tokem.fi>>
- Motiva www-sivut 2014. Hakupäivä 9.6.2014. <www.motiva.fi>
- MySQL www-sivut 2014. Hakupäivä 15.5.2014.
<<http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html>>
- Ohjelmointiputka www-sivut 2009, MySQL ja PHP. Hakupäivä 20.6.2014.
<<http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=mysqlphp01/>>
- pChart www-sivut 2014. pChart Online documentation. Hakupäivä 23.6.2014.
<<http://www.pchart.net>>
- pChart www-sivut 2014. Features. Hakupäivä 23.6.2014.
<<http://www.pchart.net/features-data>>
- pChart Sourceforge www-sivut 2014. Hakupäivä 10.5.2014.
<<http://pchart.sourceforge.net/documentation.php>>
- pChart wiki www-sivut 2014. Hakupäivä 10.5.2014.
<<http://wiki.pchart.net/doc/mysql.integration.html>>
- PHP-manual www-sivut 2014. Hakupäivä 15.5.2014.
<<http://www.php.net/manual/en/datetime.format.php>>
- php-perusteet www-sivut 2009. Hakupäivä 20.6.2014.
<<http://www.php-perusteet.com/>>
- Schneider-electric www-sivut, KNX Basic weather station (SE_NO_FI_DA).pdf. Hakupäivä 1.7.2014.
<<http://ecatalogue.schneider-electric.fi/Product-Sheet.aspx?productId=850321&groupid=298059&navid=32125&navoption=1>>
- Sourceforge www-sivut 2014. Hakupäivä 29.7.2014.
<<http://sourceforge.net/projects/calibri/>>
- Suntekno Oy 2014. Aurinkopaneelit.pdf. Hakupäivä 9.6.2014.
<<http://suntekno.bonsait.fi/resources/public/tietopankki/paneelit.pdf>>

LIITTEET

- Liite 1. Splinechart_Yesterday, lähdekoodi 1
- Liite 2. BarChart_Yesterday, lähdekoodi 2
- Liite 3. BarChart_Week, lähdekoodi 3
- Liite 4. BarChart_Month_Current, lähdekoodi 4
- Liite 5. BarChart_Month_Last, lähdekoodi 5
- Liite 6. BarChart_Month_last2, lähdekoodi 6
- Liite 7. BarChart_Month_last3, lähdekoodi 7
- Liite 8. BarChart_Year, lähdekoodi 8
- Liite 9. BarChart_AllYears, lähdekoodi 9
- Liite 10. Statistic, lähdekoodi 10
- Liite 11. Statistic2, lähdekoodi 11
- Liite 12. PHP-sivujen sijaintitiedot serverillä

Liite 1 1/(2)

```

Splinechart_Yesterday.php
1 <?php
2 /* SplineChart Yesterdays Power */
3
4 ini_set('memory_limit','640M');
5
6 /* pChart library inclusions */
7 include("../pChart/class/pData.class.php");
8 include("../pChart/class/pDraw.class.php");
9 include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24
25 /* Build the query that will returns the data to graph */
26 $result = mysql_unbuffered_query('SELECT date,time,`Power [kW]` FROM pvs300
WHERE `date` = (SELECT MAX(`date`) - INTERVAL 1 DAY FROM pvs300)');
27
28 if (!$result) {
29     die('Could not query:' . mysql_error());
30 }
31
32 while($row = mysql_fetch_array($result))
33 {
34     /* Push the results of the query in an array */
35     $Date = $row["date"];
36     if ($row["Power [kW]"] <= 0) {
37         $Power[] = 0;
38     } else {
39         $Power[] = $row["Power [kW]"];
40     }
41 }
42
43 /* Save the data in the pData array */
44 $MyData->addPoints($Power,"Power");
45 $MyData->setAxisName(0,"Power");
46 $MyData->setAxisUnit(0,"kW");
47 $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);

```

Liite 1 2/(2)

```

48 $MyData->addPoints(array("00:00","", "", "", "", "", "01:00","", "", "", "", "", "02:00","", "",
    "", "", "", "03:00","", "", "", "", "", "04:00","", "", "", "", "", "05:00","", "", "", "",
    "", "06:00","", "", "", "", "", "07:00","", "", "", "", "", "08:00","", "", "", "", "",
    "09:00","", "", "", "", "", "10:00","", "", "", "", "", "11:00","", "", "", "", "",
    "12:00","", "", "", "", "", "13:00","", "", "", "", "", "14:00","", "", "", "", "",
    "15:00","", "", "", "", "", "16:00","", "", "", "", "", "17:00","", "", "", "", "",
    "18:00","", "", "", "", "", "19:00","", "", "", "", "", "20:00","", "", "", "", "",
    "21:00","", "", "", "", "", "22:00","", "", "", "", "", "23:00","", "", "", "", ""), "Labels");
49 $MyData->setSerieDescription("Labels", "Times");
50 $MyData->setAbscissa("Labels");
51 $MyData->setPalette("Power", array("R"=>55, "G"=>91, "B"=>127));
52
53 /* Create the pChart object */
54 $myPicture = new pImage(1905,1065,$MyData);
55 /* Pure grey background*/
56 //
57 $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Star
    tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
58 /* Clouds background */
59 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
60 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
61
62 /* Write the picture title */
63 $myPicture->setFontProperties(array("FontName"=>".\pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
64 $myPicture->drawText(210,175,"Power output of the solar system during $Date",array(
    "FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
65
66 /* Do some cosmetic and draw the chart */
67 $myPicture->setGraphArea(210,180,1720,820);
68 $myPicture->drawFilledRectangle(210,180,1720,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
69 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
70
71 /* Draw a spline chart on top */
72 $myPicture->setFontProperties(array("FontName"=>".\pChart/fonts/pf_arma_five.ttf",
    "FontSize"=>14));
73 $myPicture->drawFilledSplineChart();
74
75 $myPicture->setShadow(TRUE,array("X"=>2,"Y"=>2,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
76 $myPicture->drawSplineChart();
77 $myPicture->setShadow(FALSE);
78
79 /* Render the picture (choose the best way) */
80 $myPicture->autoOutput("pictures/Day_Power.png");
81
82 unset($x);
83 unset($result);
84 unset($PowerHour);
85 unset($Date);
86 unset($MyData);
87 unset($myPicture);
88
89 ?>

```

Liite 2 1/(2)

```

BarChart_Yesterday.php
1 <?php
2 /* SplineChart Yesterdays Power */
3
4 ini_set('memory_limit','650M'); // Memory limit is now gone
5
6 /* pChart library inclusions */
7 include("../pChart/class/pData.class.php");
8 include("../pChart/class/pDraw.class.php");
9 include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24
25 /* Build the query that will returns the data to graph */
26 $variable = $_GET[x];
27 for ($x = 0; $x <= 23; $x++) { // Count last week days to get data there
28     $result = mysql_unbuffered_query("SELECT date,time,SUM(`Power [kW]`) FROM pvs300
    WHERE DATE(date) = DATE(NOW() - INTERVAL 1 DAY) AND HOUR(time) = '$x'"); // eilisen
    päivän tiedot viimeiseltä 24 tunnilta
29
30 if (!$result) {
31     die('Could not query: ' . mysql_error()); }
32
33 /* Push the results of the query in an array */
34 while($row = mysql_fetch_array($result)) {
35     $Date = $row["date"];
36     $PowerHour[] = $row["SUM(`Power [kW]`)"]; }
37 }
38
39 /* Count energy per hour and set it in right form */
40 for ($x=0 ; $x <= 23; $x++) {
41     $PowerHour[$x] = $PowerHour[$x] / 6;
42     $PowerHour[$x] = Round($PowerHour[$x], 2);
43     if ($PowerHour[$x] < 0) {
44         $PowerHour[$x] = 0; }
45 }
46
47 /* Save the data in the pData array */
48 $MyData->addPoints($PowerHour,"Power");
49 $MyData->setAxisName(0,"Power/Hour");
50 $MyData->setAxisUnit(0,"kWh"); // set y-axis unit

```

Liite 2 2/(2)

```

51 $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
52 $MyData->addPoints(array("00:00","01:00","02:00","03:00","04:00","05:00","06:00",
    "07:00","08:00","09:00","10:00","11:00","12:00","13:00","14:00","15:00","16:00",
    "17:00","18:00","19:00","20:00","21:00","22:00","23:00"), "Labels");
53 $MyData->setSerieDescription("Labels", "Times");
54 $MyData->setAbscissa("Labels");
55 $MyData->setPalette("Power", array("R"=>55, "G"=>91, "B"=>127));
56
57 /* Create the pChart object */
58 $myPicture = new pImage(1905,1065,$MyData);
59 /* Pure grey background*/
60 //
$myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"StartG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
61 /* Clouds background */
62 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
63 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
64
65 /* Write the picture title */
66 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
67 $myPicture->drawText(210,175,"Energy production of the solar system during $Date",
    array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
68
69 /* Do some cosmetic and draw the chart */
70 $myPicture->setGraphArea(210,180,1720,820);
71 $myPicture->drawFilledRectangle(210,180,1720,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
72 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
73
74 /* Draw the scale and the 1st chart */
75 $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
76 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>14));
77 $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
    "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
78 $myPicture->setShadow(FALSE);
79
80 /* Render the picture (choose the best way) */
81 $myPicture->autoOutput("pictures/Day_kWh.png");
82
83 unset($x);
84 unset($result);
85 unset($PowerHour);
86 unset($Date);
87 unset($MyData);
88 unset($myPicture);
89 ?>

```

Liite 3 1/(2)

```

BarChart_Week.php
1  k?php
2  /* Barchart Week Energy Production */
3
4  ini_set('memory_limit','640M'); //
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Build the query that will returns the data to graph */
25 $variable = $_GET[x];
26 for ($x = 0; $x <= 6; $x++) { // Count last week days to get data there
27     $result = mysql_unbuffered_query("SELECT WEEK(date),date,SUM(`Power [kW]`) FROM
pv300 WHERE WEEK(date,1) = WEEK(NOW() - INTERVAL 0 DAY) AND WEEKDAY(date) = '$x'");
    // Seitsemän päivän tiedot viimeiseltä viikolta
28
29     if (!$result) {
30         die('Could not query: ' . mysql_error());
31     }
32     /* Push the results of the query in an array */
33     while($row = mysql_fetch_array($result)) {
34         $PowerDay[] = $row["SUM(`Power [kW]`)"]; // Sum energy production per day
35         $Week = $row["WEEK(date)"]; // Week Number
36         $Date[] = $row["date"]; } // Set dates mon to sun
37     }
38
39     /* Counts daily energy in right form and round it */
40     for ($x = 0; $x <= 6; $x++) {
41         $PowerDay[$x] = $PowerDay[$x] / 6;
42         $PowerDay[$x] = Round($PowerDay[$x], 1);
43         if ($PowerDay[$x] < 0) {
44             $PowerDay[$x] = 0; }
45     }
46
47     /* Save the data in the pData array */
48     $MyData->addPoints($PowerDay,"Power"); // adds energy production per day data in
Mydata
49     $MyData->setAxisName(0,"Energy/Day"); // y-axis name
50     $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
51     $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);

```

Liite 3 2/(2)

```

52 $MyData->addPoints(array("Mon $Date[0]","Tue $Date[1]","Wen $Date[2]","Thu $Date[3]",
    "Fri $Date[4]","Sat $Date[5]","Sun $Date[6]"), "Labels"); // X-axis labels days and
    dates
53 $MyData->setSerieDescription("Labels", "Times");
54 $MyData->setAbscissa("Labels");
55 $MyData->setPalette("Power", array("R"=>55, "G"=>91, "B"=>127));
56
57 /* Create the pChart object */
58 $myPicture = new pImage(1905,1065,$MyData);
59 /* Pure grey background*/
60 //
    $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Star
    tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
61 /* Clouds background */
62 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
63 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
64
65 /* Write the picture title */
66 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
67 $myPicture->drawText(110,175,"Energy production of the solar system during week $Week
    ",array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
68
69 /* Do some cosmetic and draw the chart */
70 $myPicture->setGraphArea(110,180,1820,820);
71 $myPicture->drawFilledRectangle(110,180,1820,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
72 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
73
74 /* Draw the scale and the 1st chart */
75 $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
76 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>14));
77 $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
    "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
78 $myPicture->setShadow(FALSE);
79
80 /* Render the picture (choose the best way) */
81 $myPicture->autoOutput("pictures/Week_productionkWh.png");
82
83 unset($x);
84 unset($result);
85 unset($Week);
86 unset($PowerDay);
87 unset($Date);
88 unset($MyData);
89 unset($myPicture);
90 ?>

```


Liite 4 1/(2)

```

BarChart_Month_Current.php
1  <?php
2  /* Barchart Months Energy Production */
3
4  ini_set('memory_limit','640M'); //
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Get last date and last month name from database */
25 $result = mysql_unbuffered_query("SELECT date,MONTHNAME(date) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW())");
26 while($row = mysql_fetch_array($result)) {
27     $Date = $row["date"]; // Latest date in last month
28     $Month = $row["MONTHNAME(date)"]; // Month name
29     $daymax = date("t", strtotime($Date)); // Determine amount of days in last month
30
31     /* Build the query that will returns the data to graph */
32     $variable = $_GET[x];
33     for ($x = 1; $x <= $daymax; $x++) { // Count last month days to get data there
34         $result = mysql_unbuffered_query("SELECT SUM(`Power [kW]`) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW()) AND DAY(date) = '$x'");
35
36         if (!$result) {
37             die('Could not query: ' . mysql_error());
38         }
39         /* Push the results of the query in an array */
40         while($row = mysql_fetch_array($result)) {
41             $PowerDay[] = $row["SUM(`Power [kW]`)"];
42         }
43
44         /* Counts Daily energy in right form and round it */
45         for ($x = 0; $x <= $daymax - 1; $x++) {
46             $PowerDay[$x] = $PowerDay[$x] / 6;
47             $PowerDay[$x] = Round($PowerDay[$x], 1);
48             if ($PowerDay[$x] < 0) {
49                 $PowerDay[$x] = 0;
50             }
51         }
52
53         /* Save the data in the pData array */
54         $MyData->addPoints($PowerDay,"Power"); // adds power data in Mydata

```


Liite 4 2/(2)

```

54 $MyData->setAxisName(0,"Energy/Day"); // y-axis name
55 $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
56 $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
57 for ($day = 1; $day <= $daymax; $day++) { // Count last month days to create x-axis
    labels
58 $MyData->addPoints(array($day),"Labels"); }
59 $MyData->setSerieDescription("Labels","Times");
60 $MyData->setAbscissa("Labels");
61 $MyData->setPalette("Power",array("R"=>55,"G"=>91,"B"=>127));
62
63 /* Create the pChart object */
64 $myPicture = new pImage(1905,1065,$MyData);
65 /* Pure grey background*/
66 //
67 $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Star
    tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
68 /* Clouds background */
69 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
70 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
71
72 /* Write the picture title */
73 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
74 $myPicture->drawText(110,175,"Energy production of the solar system during $Month ",
    array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
75
76 /* Do some cosmetic and draw the chart */
77 $myPicture->setGraphArea(110,180,1820,820);
78 $myPicture->drawFilledRectangle(110,180,1820,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
79 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
80
81 /* Draw the scale and the 1st chart */
82 $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
83 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>14));
84 $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
    "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
85 $myPicture->setShadow(FALSE);
86
87 /* Render the picture (choose the best way) */
88 $myPicture->autoOutput("pictures/Month_productionkWh.png");
89
90 unset($x);
91 unset($day);
92 unset($result);
93 unset($row);
94 unset($Month);
95 unset($daymax);
96 unset($PowerDay);
97 unset($Date);
98 unset($MyData);
99 unset($myPicture);
100 ?>

```

Liite 5 1/(2)

```

BarChart_Month_Last.php
1  <?php
2  /* Barchart Months Energy Production */
3
4  ini_set('memory_limit','640M'); // Memory limit is now gone
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Get last date and last month name from database */
25 $result = mysql_unbuffered_query("SELECT date,MONTHNAME(date) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW() - INTERVAL 1 Month)");
26 while($row = mysql_fetch_array($result)) {
27     $Date = $row["date"]; // Latest date in last month
28     $Month = $row["MONTHNAME(date)"]; // Month name
29     $daymax = date("t", strtotime($Date)); // Determine amount of days in last month
30
31     /* Build the query that will returns the data to graph */
32     $variable = $_GET[x];
33     for ($x = 1; $x <= $daymax; $x++) { // Count last month days to get data there
34         $result = mysql_unbuffered_query("SELECT SUM(`Power [kw]`) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW() - INTERVAL 1 Month) AND DAY(date) = '$x'");
35
36         if (!$result) {
37             die('Could not query: ' . mysql_error());
38         }
39         /* Push the results of the query in an array */
40         while($row = mysql_fetch_array($result)) {
41             $PowerDay[] = $row["SUM(`Power [kw]`)"];
42         }
43
44         /* Counts daily energy in right form and round it */
45         for ($x = 0; $x <= $daymax - 1; $x++) {
46             $PowerDay[$x] = $PowerDay[$x] / 6;
47             $PowerDay[$x] = Round($PowerDay[$x], 1);
48             if ($PowerDay[$x] < 0) {
49                 $PowerDay[$x] = 0;
50             }
51         }
52         /* Save the data in the pData array */
53         $MyData->addPoints($PowerDay,"Power"); // adds power data in Mydata

```

Liite 5 2/(2)

```

54 $MyData->setAxisName(0,"Energy/Day"); // y-axis name
55 $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
56 $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
57 for ($day = 1; $day <= $daymax; $day++) { // Count last month days to create x-axis
    labels
58 $MyData->addPoints(array($day),"Labels"); }
59 $MyData->setSerieDescription("Labels","Times");
60 $MyData->setAbscissa("Labels");
61 $MyData->setPalette("Power",array("R"=>55,"G"=>91,"B"=>127));
62
63 /* Create the pChart object */
64 $myPicture = new pImage(1905,1065,$MyData);
65 /* Pure grey background*/
66 //
67 $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Start
    tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
68 /* Clouds background */
69 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
70 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
71
72 /* Write the picture title */
73 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
74 $myPicture->drawText(110,175,"Energy production of the solar system during $Month ",
    array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
75
76 /* Do some cosmetic and draw the chart */
77 $myPicture->setGraphArea(110,180,1820,820);
78 $myPicture->drawFilledRectangle(110,180,1820,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
79 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
80
81 /* Draw the scale and the 1st chart */
82 $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
83 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>14));
84 $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
    "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
85 $myPicture->setShadow(FALSE);
86
87 /* Render the picture (choose the best way) */
88 $myPicture->autoOutput("pictures/Month_productionkWh.png");
89
90 unset($x);
91 unset($day);
92 unset($result);
93 unset($row);
94 unset($Month);
95 unset($daymax);
96 unset($PowerDay);
97 unset($Date);
98 unset($MyData);
99 unset($myPicture);
100 ?>

```

Liite 6 1/(2)

```

BarChart_Month_last2.php
1  <?php
2  /* Barchart Months Energy Production */
3
4  ini_set('memory_limit','640M'); // Memory limit is now gone
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Get last date and last month name from database */
25 $result = mysql_unbuffered_query("SELECT date,MONTHNAME(date) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW() - INTERVAL 2 Month)");
26 while($row = mysql_fetch_array($result)) {
27     $Date = $row["date"]; // Latest date in last month
28     $Month = $row["MONTHNAME(date)"]; // Month name
29     $daymax = date("t", strtotime($Date)); // Determine amount of days in last month
30
31     /* Build the query that will returns the data to graph */
32     $variable = $_GET[x];
33     for ($x = 1; $x <= $daymax; $x++) { // Count last month days to get data there
34         $result = mysql_unbuffered_query("SELECT SUM(`Power [kW]`) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW() - INTERVAL 2 Month) AND DAY(date) = '$x'");
35
36         if (!$result) {
37             die('Could not query: ' . mysql_error());
38         }
39
40         /* Push the results of the query in an array */
41         while($row = mysql_fetch_array($result)) {
42             $PowerDay[] = $row["SUM(`Power [kW]`)"];
43         }
44
45         /* Counts daily energy in right form and round it */
46         for ($x = 0; $x <= $daymax - 1; $x++) {
47             $PowerDay[$x] = $PowerDay[$x] / 6;
48             $PowerDay[$x] = Round($PowerDay[$x], 1);
49             if ($PowerDay[$x] < 0) {
50                 $PowerDay[$x] = 0;
51             }
52         }
53
54         /* Save the data in the pData array */
55         $MyData->addPoints($PowerDay,"Power"); // adds power data in Mydata

```

Liite 6 2/(2)

```

54 $MyData->setAxisName(0,"Energy/Day"); // y-axis name
55 $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
56 $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
57 for ($day = 1; $day <= $daymax; $day++) { // Count last month days to create x-axis
    labels
58 $MyData->addPoints(array($day),"Labels"); }
59 $MyData->setSerieDescription("Labels","Times");
60 $MyData->setAbscissa("Labels");
61 $MyData->setPalette("Power",array("R"=>55,"G"=>91,"B"=>127));
62
63 /* Create the pChart object */
64 $myPicture = new pImage(1905,1065,$MyData);
65 /* Pure grey background*/
66 //
67 $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"StartG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
68 /* Clouds background */
69 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
70 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
71
72 /* Write the picture title */
73 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
74 $myPicture->drawText(110,175,"Energy production of the solar system during $Month ",
    array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
75
76 /* Do some cosmetic and draw the chart */
77 $myPicture->setGraphArea(110,180,1820,820);
78 $myPicture->drawFilledRectangle(110,180,1820,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
79 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
80
81 /* Draw the scale and the 1st chart */
82 $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
83 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>14));
84 $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
    "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
85 $myPicture->setShadow(FALSE);
86
87 /* Render the picture (choose the best way) */
88 $myPicture->autoOutput("pictures/Month_productionkWh.png");
89
90 unset($x);
91 unset($day);
92 unset($result);
93 unset($row);
94 unset($Month);
95 unset($daymax);
96 unset($PowerDay);
97 unset($Date);
98 unset($MyData);
99 unset($myPicture);
100
101 ?>

```

Liite 7 1/(2)

```

BarChart_Month_last3.php
1  <?php
2  /* Barchart Months Energy Production */
3
4  ini_set('memory_limit','640M'); // Memory limit is now gone
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Get last date and last month name from database */
25 $result = mysql_unbuffered_query("SELECT date,MONTHNAME(date) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW() - INTERVAL 3 Month)");
26 while($row = mysql_fetch_array($result)) {
27     $Date = $row["date"]; // Latest date in last month
28     $Month = $row["MONTHNAME(date)"]; // Month name
29     $daymax = date("t", strtotime($Date)); // Determine amount of days in last month
30
31     /* Build the query that will returns the data to graph */
32     $variable = $_GET[x];
33     for ($x = 1; $x <= $daymax; $x++) { // Count last month days to get data there
34         $result = mysql_unbuffered_query("SELECT SUM(`Power [kW]`) FROM pvs300 WHERE
MONTH(date) = MONTH(NOW() - INTERVAL 3 Month) AND DAY(date) = '$x'");
35
36         if (!$result) {
37             die('Could not query:' . mysql_error()); }
38
39         /* Push the results of the query in an array */
40         while($row = mysql_fetch_array($result)) {
41             $PowerDay[] = $row["SUM(`Power [kW]`)"]; }
42         }
43
44         /* Counts daily energy in right form and round it */
45         for ($x = 0; $x <= $daymax - 1; $x++) {
46             $PowerDay[$x] = $PowerDay[$x] / 6;
47             $PowerDay[$x] = Round($PowerDay[$x], 1);
48             if ($PowerDay[$x] < 0) {
49                 $PowerDay[$x] = 0; }
50         }
51
52         /* Save the data in the pData array */
53         $MyData->addPoints($PowerDay,"Power"); // adds power data in Mydata

```


Liite 7 2/(2)

```

54 $MyData->setAxisName(0,"Energy/Day"); // y-axis name
55 $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
56 $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
57 for ($day = 1; $day <= $daymax; $day++) { // Count last month days to create x-axis
    labels
58 $MyData->addPoints(array($day),"Labels"); }
59 $MyData->setSerieDescription("Labels","Times");
60 $MyData->setAbscissa("Labels");
61 $MyData->setPalette("Power",array("R"=>55,"G"=>91,"B"=>127));
62
63 /* Create the pChart object */
64 $myPicture = new pImage(1905,1065,$MyData);
65 /* Pure grey background*/
66 //
67 $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Star
    tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
68 /* Clouds background */
69 $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
70 $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
71
72 /* Write the picture title */
73 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>20));
74 $myPicture->drawText(110,175,"Energy production of the solar system during $Month ",
    array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
75
76 /* Do some cosmetic and draw the chart */
77 $myPicture->setGraphArea(110,180,1820,820);
78 $myPicture->drawFilledRectangle(110,180,1820,820,array("R"=>210,"G"=>210,"B"=>210,
    "Surrounding"=>-200,"Alpha"=>50));
79 $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
    SCALE_MODE_START0));
80
81 /* Draw the scale and the 1st chart */
82 $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
83 $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
    "FontSize"=>14));
84 $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
    "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
85 $myPicture->setShadow(FALSE);
86
87 /* Render the picture (choose the best way) */
88 $myPicture->autoOutput("pictures/Month_productionkWh.png");
89
90 unset($x);
91 unset($day);
92 unset($result);
93 unset($row);
94 unset($Month);
95 unset($daymax);
96 unset($PowerDay);
97 unset($Date);
98 unset($MyData);
99 unset($myPicture);
100 ?>

```

Liite 8 1/(2)

```

BarChart_Year.php
1  <?php
2  /* Barchart Year Energy Production version 2 */
3
4  ini_set('memory_limit','640M'); // Memory limit is now gone
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Build the query that will returns the data to graph */
25 $variable = $_GET[x];
26 for ($x = 11; $x >= 0; $x--) { // Count current year months to get data there
27     $result = mysql_unbuffered_query("SELECT date,MONTHNAME(date),SUM(`Power [kW]`) FROM
pv300 WHERE MONTH(date) = MONTH(NOW() - INTERVAL '$x' Month)");
28     if (!$result) {
29         die('Could not query: ' . mysql_error());
30     }
31     /* Push the results of the query in an array */
32     while($row = mysql_fetch_array($result)) {
33         $Month[] = $row["MONTHNAME(date)"]; // Month name
34         $PowerMonth[] = $row["SUM(`Power [kW]`)"]; // put monthly energy in variable
35     }
36
37     /* Counts monthly energy in right form and round it */
38     for ($x = 0; $x <= 11; $x++) {
39         $PowerMonth[$x] = $PowerMonth[$x] / 6;
40         $PowerMonth[$x] = Round($PowerMonth[$x], 0);
41         if ($PowerMonth[$x] < 0) {
42             $PowerMonth[$x] = 0;
43         }
44     }
45
46     /* Save the data in the pData array */
47     $MyData->addPoints($PowerMonth,"Power"); // adds power data in Mydata
48     $MyData->setAxisName(0,"Energy/Month"); // y-axis name
49     $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
50     $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
51     for ($x = 0; $x <= 11; $x++) { // Counts month to create x-axis labels
52         $MyData->addPoints(array($Month[$x]),"Labels");
53     }
54     $MyData->setSerieDescription("Labels","Month");
55     $MyData->setAbscissa("Labels");
56     $MyData->setPalette("Power",array("R"=>55,"G"=>91,"B"=>127));

```


Liite 8 2/(2)

```

55
56  /* Create the pChart object */
57  $myPicture = new pImage(1905,1065,$MyData);
58  /* Pure grey background*/
59  //
60  $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Star
61  tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
62  /* Clouds background */
63  $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
64  $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
65
66  /* Write the picture title */
67  $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
68  "FontSize"=>20));
69  $myPicture->drawText(110,175,"Energy production of the solar system during last 12
70  months",array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
71
72  /* Do some cosmetic and draw the chart */
73  $myPicture->setGraphArea(135,180,1795,820);
74  $myPicture->drawFilledRectangle(135,180,1795,820,array("R"=>210,"G"=>210,"B"=>210,
75  "Surrounding"=>-200,"Alpha"=>50));
76  $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
77  SCALE_MODE_START0));
78
79  /* Draw the scale and the 1st chart */
80  $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
81  $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
82  "FontSize"=>14));
83  $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
84  "Interleave"=>0.005,"Rounded"=>TRUE,"Surrounding"=>60));
85  $myPicture->setShadow(FALSE);
86
87  /* Render the picture (choose the best way) */
88  $myPicture->autoOutput("pictures/Year_kWh.png");
89
90  unset($x);
91  unset($result);
92  unset($PowerMonth);
93  unset($Month);
94  unset($MyData);
95  unset($myPicture);
96
97  ?>

```

Liite 9 1/(2)

```

BarChart_AllYears.php
1 <?php
2 /* Barchart Year energy production */
3
4 ini_set('memory_limit','640M'); // Memory limit is now gone
5
6 /* pChart library inclusions */
7 include("../pChart/class/pData.class.php");
8 include("../pChart/class/pDraw.class.php");
9 include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Build the query that will returns the data to graph */
25 $variable = $_GET[x];
26 for ($x = 0; $x <= 6; $x++) { // Count year to get data there
27     $result = mysql_unbuffered_query("SELECT SUM(`Power [kW]`) FROM pvs300 WHERE
YEAR(date) = YEAR(date('2014-01-01') + INTERVAL '$x' YEAR)");
28
29     if (!$result) {
30         die('Could not query: ' . mysql_error());
31     }
32
33     /* Push the results of the query in an array */
34     while($row = mysql_fetch_array($result)) {
35         $PowerYear[] = $row["SUM(`Power [kW]`)"]; } // put year energy production in variable
36     }
37
38     /* Counts year energy in right form and round it */
39     for ($x = 0; $x <= 6; $x++) {
40         $PowerYear[$x] = $PowerYear[$x] / 6;
41         $PowerYear[$x] = Round($PowerYear[$x], 0);
42         if ($PowerYear[$x] < 0) {
43             $PowerYear[$x] = 0; }
44     }
45
46     /* Save the data in the pData array */
47     $MyData->addPoints($PowerYear,"Power"); // adds power data in Mydata
48     $MyData->setAxisName(0,"Energy/Year"); // y-axis name
49     $MyData->setAxisUnit(0,"kWh"); // Y-axis unit
50     $MyData->setAxisDisplay(0,AXIS_FORMAT_DEFAULT);
51     $MyData->addPoints(array("2014","2015","2016","2017","2018","2019","2020"),"Labels");
52     $MyData->setSerieDescription("Labels","Year");
53     $MyData->setAbcissa("Labels");
54     $MyData->setPalette("Power",array("R"=>55,"G"=>91,"B"=>127));

```

Liite 9 2/(2)

```

55  /* Create the pChart object */
56  $myPicture = new pImage(1905,1065,$MyData);
57  /* Pure grey background*/
58  //
59  $myPicture->drawGradientArea(0,0,1910,1070,DIRECTION_VERTICAL,array("StartR"=>220,"Star
60  tG"=>220,"StartB"=>220,"EndR"=>255,"EndG"=>255,"EndB"=>255,"Alpha"=>100));
61  /* Clouds background */
62  $myPicture->drawFromGIF(0,0,"C:\www\srea\pilvet.gif");
63  $myPicture->drawRectangle(0,0,1904,1064,array("R"=>200,"G"=>200,"B"=>200));
64
65  /* Write the picture title */
66  $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
67  "FontSize"=>20));
68  $myPicture->drawText(135,175,"Energy production of the solar system during years
69  2014 - 2020 ",array("FontSize"=>32,"Align"=>TEXT_ALIGN_BOTTOMLEFT));
70
71  /* Do some cosmetic and draw the chart */
72  $myPicture->setGraphArea(135,180,1795,820);
73  $myPicture->drawFilledRectangle(135,180,1795,820,array("R"=>210,"G"=>210,"B"=>210,
74  "Surrounding"=>-200,"Alpha"=>50));
75  $myPicture->drawScale(array("GridR"=>180,"GridG"=>180,"GridB"=>180,"Mode"=>
76  SCALE_MODE_START0));
77
78  /* Draw the scale and the 1st chart */
79  $myPicture->setShadow(TRUE,array("X"=>1,"Y"=>1,"R"=>0,"G"=>0,"B"=>0,"Alpha"=>10));
80  $myPicture->setFontProperties(array("FontName"=>"../pChart/fonts/Forgotte.ttf",
81  "FontSize"=>14));
82  $myPicture->drawBarChart(array("DisplayValues"=>TRUE,"DisplayColor"=>DISPLAY_AUTO,
83  "Interleave"=>0.02,"Rounded"=>TRUE,"Surrounding"=>60));
84  $myPicture->setShadow(FALSE);
85
86  /* Render the picture (choose the best way) */
87  $myPicture->autoOutput("pictures/Years_kwh.png");
88
89  unset($x);
90  unset($result);
91  unset($row);
92  unset($PowerYear);
93  unset($MyData);
94  unset($myPicture);
95
96  ?>

```

Liite 10 1/(2)

```

Statistic.php
1  <?php
2  /* Statistic */
3
4  ini_set('memory_limit','640M'); // Memory limit is now gone
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24
25 /* Build the query that will returns the data to graph */
26 $result = mysql_unbuffered_query("SELECT date,`Power peak of day [kW]`,`Power peak
time of day (Unix epoch) [s]`,`Power peak ever [kW]`,`Power peak time ever (Unix
epoch) [s]`,`Total energy [kWh]` FROM pvs300 WHERE `date` = (SELECT MAX(`date`) -
INTERVAL 1 DAY FROM pvs300)");
27 if (!$result) {
28     die('Could not query: ' . mysql_error()); }
29
30 /* Push the results of the query in an array */
31 while($row = mysql_fetch_array($result)) {
32
33     $PowerDay = $row["Power peak of day [kW]"];
34     $TimePowerDay = $row["Power peak time of day (Unix epoch) [s]"];
35     $PowerBest = $row["Power peak ever [kW]"];
36     $TimePowerBest = $row["Power peak time ever (Unix epoch) [s]"];
37     $TotalEnergy = $row["Total energy [kWh]"]; }
38
39 $result = mysql_unbuffered_query("SELECT date,`Power [kW]`,`Total energy [kWh]` FROM
pvs300 WHERE `date` = (SELECT MAX(`date`) FROM pvs300)");
40 if (!$result) {
41     die('Could not query: ' . mysql_error()); }
42
43 /* Push the results of the query in an array */
44 while($row = mysql_fetch_array($result)) {
45     $CurrentPower = $row["Power [kW]"];
46     $TotalEnergy = $row["Total energy [kWh]"];
47     $Date = $row["date"];
48 }
49
50 /* Date and Energy variables to correct format */
51 $TimeDay = date("d/F/Y H:i", $TimePowerDay);

```

Liite 10 2/(2)

```

52 $DateBest = date("d/F/Y H:i", $TimePowerBest);
53 $TotalEnergyDate = date("d/F/Y", strtotime($Date));
54 $TotalEnergy = Round($TotalEnergy, 0);
55
56 /* Draw visible page */
57 // echo "<body background='/srea/sitra_blog_solar_abb_imagebank2.jpg'>";
58 echo "<body background='/srea/pilvet.gif'>";
59 echo "<font size='18' face='Arial'>";
60     echo "<br><br>";
61     echo "<Center>Solar system statistics<br><br><br>";
62     echo "Total energy output $TotalEnergy kWh<br>";
63     echo "$TotalEnergyDate<br><br>";
64     echo "The current power output $CurrentPower kW<br><br>";
65     echo "Yesterday's highest power peak $PowerDay kW<br>";
66     echo "$TimeDay<br><br>";
67     echo "The highest power peak ever $PowerBest kW<br>";
68     echo "$DateBest<br><br>";
69
70 unset($PowerDay);
71 unset($TimePowerDay);
72 unset($result);
73 unset($row);
74 unset($PowerBest);
75 unset($TimePowerBest);
76 unset($TotalEnergy);
77 unset($CurrentPower);
78 unset($Date);
79 unset($TimeDay);
80 unset($DateBest);
81 unset($TotalEnergyDate);
82 unset($MyData);
83 unset($myPicture);
84 ?>

```

Liite 11 1/(2)

```

Statistic2.php
1  k?php
2  /* Statistics2 */
3
4  ini_set('memory_limit','640M'); // Memory limit is now gone
5
6  /* pChart library inclusions */
7  include("../pChart/class/pData.class.php");
8  include("../pChart/class/pDraw.class.php");
9  include("../pChart/class/pImage.class.php");
10
11 /* Create and populate the pData object */
12 $MyData = new pData();
13
14 /* Connect to the MySQL database */
15 $db = mysql_connect("localhost", "token", "token");
16 if (!$db) {
17     die('Could not connect: ' . mysql_error());
18 }
19 if (!mysql_select_db('srea_data')) {
20     die('Could not select database: ' . mysql_error());
21 }
22 mysql_select_db("srea_data",$db);
23
24 /* Build the query that will returns the data to graph 22*/
25 $variable = $_GET[x];
26 for ($x = 1; $x <= 12; $x++) { // Count current year months to get data there
27     $result = mysql_unbuffered_query("SELECT date, YEAR(CURDATE()), SUM(`Power [kw]`), MONTH(CURDATE())
FROM pvs300 WHERE YEAR(date) = YEAR(CURDATE()) AND MONTH(date) = '$x'");
28
29     if (!$result) {
30         die('Could not query: ' . mysql_error());
31     }
32
33     /* Push the results of the query in an array */
34     while($row = mysql_fetch_array($result)) {
35         $CurrentMonth = $row["MONTH(CURDATE())"];
36         $DateMonth[] = $row["date"];
37         $Year = $row["YEAR(CURDATE())"];
38         $PowerMonth[] = $row["SUM(`Power [kw]`)"];
39     }
40
41     /* Build the query that will returns the data to graph */
42     $variable = $_GET[x];
43     for ($x = 1; $x <= 366; $x++) { // Count current year months to get data there
44         $result = mysql_unbuffered_query("SELECT date, SUM(`Power [kw]`) FROM pvs300 WHERE YEAR(date) =
YEAR(CURDATE()) AND DAYOFYEAR(date) = '$x'");
45
46         if (!$result) {
47             die('Could not query: ' . mysql_error());
48         }
49
50         /* Push the results of the query in an array */
51         while($row = mysql_fetch_array($result)) {
52             $DateDay[] = $row["date"];
53             $PowerDay[] = $row["SUM(`Power [kw]`)"];
54         }
55     }
56 }

```

Liite 11 2/(2)

```

54
55 /* Lines to find Best and Worst output months */
56 for ($y = 0; $y <= 11; $y++) {
57     if ($PowerMonth[$y] > $PowerMonthBest) {
58         $PowerMonthBest = $PowerMonth[$y];
59         $DateMonthBest = $DateMonth[$y]; }
60     }
61     $PowerMonthWorst = $PowerMonthBest;
62     for ($y = 0; $y <= 11; $y++) {
63         if ($PowerMonth[$y] <> 0 AND $PowerMonth[$y] < $PowerMonthWorst AND $CurrentMonth <> $y+1) {
64             $PowerMonthWorst = $PowerMonth[$y];
65             $DateMonthWorst = $DateMonth[$y]; }
66     }
67
68 /* Lines to find Best output day and its date */
69 for ($y = 0; $y <= 365; $y++) {
70     if ($PowerDay[$y] > $PowerDayBest) {
71         $PowerDayBest = $PowerDay[$y];
72         $DateDayBest = $DateDay[$y]; }
73     }
74
75 /* Lines of Variable design, rounding and outlooks */
76 $DateDayBest = date("d/F", strtotime($DateDayBest));
77 $DateMonthBest = date("F", strtotime($DateMonthBest));
78 $DateMonthWorst = date("F", strtotime($DateMonthWorst));
79 $PowerMonthBest = $PowerMonthBest / 6;
80 $PowerMonthBest = Round($PowerMonthBest, 0);
81 $PowerMonthWorst = $PowerMonthWorst / 6;
82 $PowerMonthWorst = Round($PowerMonthWorst, 0);
83 if ($PowerMonthWorst[$x] < 0) {
84     $PowerMonthWorst[$x] = 0; }
85 $PowerDayBest = $PowerDayBest / 6;
86 $PowerDayBest = Round($PowerDayBest, 0);
87
88 /* Draw visible page */
89 // echo "<body background='/srea/sitra_blog_solar_abb_imagebank2.jpg'>";
90 echo "<body background='/srea/pilvet.gif'>";
91 echo "<font size='24' face='Arial'>";
92     echo "<br><br><br>";
93     echo "<Center>Solar system statistics in year $Year<br><br><br>";
94     echo "Best production day in this year is $DateDayBest<br>";
95     echo "$PowerDayBest kWh<br><br>";
96     echo "Best production month in this year is $DateMonthBest<br>";
97
98     echo "$PowerMonthBest kWh<br><br>";
99     echo "Worst production month in this year is $DateMonthWorst<br>";
100     echo "$PowerMonthWorst kWh<br><br>";
101
102 unset($x);
103 unset($y);
104 unset($result);
105 unset($row);
106 unset($CurrentMonth);
107 unset($DateMonth);
108 unset($Year);
109 unset($PowerMonth);
110 unset($MyData);
111 unset($myPicture);
112 unset($DateDay);
113 unset($PowerDay);
114 unset($result);
115 unset($row);
116 unset($PowerMonthBest);
117 unset($DateMonthBest);
118 unset($PowerMonthWorst);
119 unset($DateMonthWorst);
120 unset($MyData);
121 unset($myPicture);
122
123

```

Liite 12 1/(2)

Opinnäytetyö

25.8.2014

Petri Juntunen

Aurinkopaneeleiden tehon seurannan php sivujen sijainti serverikoneella. Nämä Infojärjestelmässä olevat sivut pitäisi saada näkymään koulun Internet -sivuilla.

Serverin IP: <http://172.19.131.6>

Alla on sijainti sivulle, joka näyttää aurinkopaneeleiden eilisen päivän tehon seurannan käyrä muodossa.

C:\www\Valmiit\Splinechart_Yesterday.php

Alla on sijainti sivulle, joka näyttää aurinkopaneeleiden eilinen energian tuoton pylväs muodossa.

C:\www\Valmiit\BarChart_Yesterday.php

Alla on sijainti sivulle, joka näyttää aurinkopaneeleiden edellisen viikon energian tuoton pylväs muodossa.

C:\www\Valmiit\BarChart_Week.php

Alla on sijainti sivulle, joka näyttää aurinkopaneeleiden kuluvan kuukauden energian tuoton pylväs muodossa.

C:\www\Valmiit\BarChart_Month_Current.php

Alla on sijainnit kolmelle sivulle, jotka näyttää energian tuoton edellisiltä kolmelta kuukaudelta pylväs muodossa.

C:\www\Valmiit\BarChart_Month_Last.php

C:\www\Valmiit\BarChart_Month_Last2.php

C:\www\Valmiit\BarChart_Month_Last3.php

Alla on sijainti sivulle, joka näyttää aurinkopaneeleiden kuluvan vuoden energian tuoton joka kuukaudelta pylväs muodossa.

C:\www\Valmiit\BarChart_Year.php

Liite 12 2/(2)

Alla on sijainti sivulle, joka näyttää aurinkopaneelien energiantuoton vuosittain.

C:\www\Valmiit\BarChart_AllYears.php

Kaksi alla olevaa sivua näyttää tilastotietoja aurinkopaneelien toiminnasta.

C:\www\Valmiit\Statistic.php

C:\www\Valmiit\Statistic2.php

Alla olevassa powerpoint esityksessä näkyy otsikot, kuvat, vähän tietoa aurinkopaneelista ja sähköauton latauspisteestä.

C:\www\Valmiit\ Solar1.0.pptx